DROP IMPACT RELIABILITY TESTING LEAD-FREE CHIP SCALE PACKAGES

A Thesis

Presented to

the Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

by

Andrew Farris

June 2008

AUTHORIZATION FOR REPRODUCTION OF MASTER'S THESIS

I grant permission for the reproduction of this thesis in its entirety or any of its parts, without further authorization from me.

Signature

05 June 2008

Date

APPROVAL PAGE

TITLE: Drop Impact Reliability Testing Lead-free Chip Scale Packages

AUTHOR: Andrew Farris

DATE SUBMITTED: June 2008

Dr. Albert Liddicoat	Albert A. Liddicoat
Advisor or Committee Chair	Signature
Dr. Jianbaio Pan	Jianbaio Pan
Committee Member	Signature
Dr. James Harris	James A Harris
Committee Member	Signature

Abstract

Drop Impact Reliability Testing Lead-free Chip Scale Packages

by

Andrew Farris

Mobile and handheld electronics devices such as digital cameras, cell phones, and personal digital assistants (PDAs) are prone to be dropped in their lifetime. The drop event may result in failure of solder joints inside these devices. The importance and widespread use of these devices in both business and leisure activities continues to increase, so device failure is increasingly costly and inconvenient.

Recently the European Union (EU) Restriction of Hazardous Substances (RoHS) and other countries' lead-free directives banned the use of lead in consumer electronics products. While this is a responsible environmental change for the electronics industry, it requires the introduction of new solders and soldering processes, and signals a major change in production methods as lead-based solders are no longer used in these devices. Thus, it is critical to study the drop impact reliability of lead-free solder joints.

This thesis discusses the reliability of Chip Scale Packages (CSPs) in drop impact, with and without the CSPs being edge-bonded, using two failure detection systems and presents the component failure sequence as observed by each system. To enable this study a high-speed data acquisition system, capable of in-situ detection of transitional solder interconnect failure, was developed at Cal Poly for drop impact reliability testing. Edge-bonding is shown to significantly improve drop impact reliability of CSPs.

Acknowledgements

I owe my love of learning, dedication, and integrity to my upbringing. My parents taught me the value of hard work and excellence by example. Their lives are a testament to the saving grace of our Lord and Savior, and the interest they have taken in all my endeavors, as well as in my siblings, is tireless.

I am grateful for the help and support of the following fellow students and colleagues, whose combined contributions enabled the success of this multidisciplinary project: Nicholas Vickers (MATE), Micah Denecour (IE), Michael Krist (IE), Chi-Yeh Hsu (ME), Edward Clements (MFGE), and Keith Rothman (AERO). The development of the data acquisition system prototype, software development, and failure analysis was a collaborative effort and the contributions of each member of this group improved the quality of the results.

The guidance and support of the faculty of Cal Poly during my study has been paramount to my success as a student, and will play a role in my successes as an engineer. I am grateful for the opportunity to have been a part of this university's legacy of excellence in engineering education. I am thankful that my thesis advisors, Dr. Jianbaio Pan, and Dr. Albert Liddicoat, have been supportive of my work; together they have tremendously improved my presentation and technical writing skills. My thanks also goes to Dr. James Harris for his participation on my thesis committee.

This work was supported in part by the Department of the Navy, Office of Naval Research, under Award #N00014-06-1-1111. Technical guidance and support was also provided by Flextronics International and Henkel Corporation; their support of research at Cal Poly is appreciated.

Others have made contributions to this project and otherwise supported my education — my thanks goes to each of them for their effort.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview of Drop Impact Reliability

Solder reliability has been formally studied for many years, but only recently has drop impact reliability been the primary focus of so many researchers. Rapid advancements in the handheld electronics industry, combined with the recent regulated removal of lead-based solders from these products, has led researchers to seek the answers to many questions on lead-free solder drop impact reliability.

To understand the purpose of drop impact reliability testing a brief introduction to thermal reliability testing is useful. Traditionally, solder reliability was studied only in thermal shock or thermal cycling tests, for which the electronic device is subjected to a series of heating and cooling cycles in a combination of elevated temperature and humidity so that the device failure rate is accelerated. As the device heats and cools repeatedly the electrical continuity of solder interconnects in the device are monitored. Thermal cycling tests are an accepted method of accelerating the failure of an electronic device due to normal use, such as power-on and power-off operations over a several year lifespan. The matching of thermal expansion rates of the materials

used in the electronic device is one goal of thermal cycling tests.

Drop impact reliability is the study of how effectively an electronic device survives a sudden mechanical force being applied to the electronic device. The mechanical force is typically applied by dropping the device and letting it impact against a firm surface. The impact causes mechanical damage to the electronic device due to stress and strain in the materials used to fabricate it. Typically what is measured is how many times an electronic device can survive the same drop impact forces before electrical failure occurs. Characterizing solder interconnect impact failure mechanisms and improving the design of the electronic device, or the quality of the materials used to manufacture it, so as to withstand drop impact forces is the primary goal of drop impact tests.

Drop impact tests are not accelerated use tests because electronic devices are generally not intended to be repeatedly dropped during normal use. However, because handheld electronics are carried they may be dropped during their lifetime, causing drop impact related failure. Improvements in drop impact reliability are intended to keep a device working in the rare case it is dropped, rather than intended to extend its normal lifetime. The military, aerospace, automotive, and other industries are also using many electronic devices in environments that may cause mechanical damage similarly to drop impacts, so drop impact reliability testing has many applications in the electronics industry. It would be difficult to refute however, that the recent proliferation of handheld consumer electronics products has greatly increased the industry's interest in drop impact reliability.

A drop impact test is typically conducted by first attaching a test vehicle (the electronic device) on a shock test machine's drop table, then raising the drop table to a predetermined height, and then releasing the table allowing it to fall vertically until it impacts the shock test machine's base. The impact of the table slamming into the base applies a half-sine shaped acceleration pulse to the test vehicle; this pulse

results from the rapid change of velocity and is described by an acceleration peak value and pulse duration. The severity of the test is usually controlled by the drop height (higher or lower), and the impact surface (softer or harder) between the drop table and base plate. Drop impact tests are repeated until the electronic device fails due to electrical conductivity loss.

This thesis contributes to the reliability of microelectronics assemblies by identifying board laminate cohesive failure as the primary failure mode of this JEDEC standard [1] lead-free CSP drop impact test vehicle. Additionally, a significant reliability improvement is reported from edge-bonding 12mm CSPs. The acceleration conditions observed on the test vehicle for each component location are reported for a variety of test vehicle assembly states. High-speed data acquisition is validated as an effective method of failure detection for intermittent drop impact failures, but several issues arising due to the wire to board connections for this system are identified.

1.2 Previous Work

There has been a significant amount of research done in the last few years on drop impact reliability. The JEDEC standard JESD22-B111 [1] for the board level and related standards [2, 3] for subassembly level have been developed for drop testing handheld electronics. Lim and Low [4] proposed a method to examine the drop impact responses of portable electronic products at different impact orientations and drop height. The impact behavior has been studied at the product level [5]. After comprehensive drop tests, failure analysis, and simulations on two ball grid array (BGA) packages at the board level, Tee, et al. [6] developed a life prediction model for board level drop testing. The effect of different solder alloy compositions on drop reliability has been studied by Syed, et al. [7]. Since SnAgCu (SAC) solder alloy performs poorly compared with SnPb solder under drop test, several studies have

been done to improve the reliability of lead-free solder joints by adding micro-alloying additions [8, 9] or lowering Ag content [10].

Underfill materials were originally developed to improve the solder joint reliability of ball-grid array (BGA) and flip chip packages during temperature cycling. Recently studies have shown that underfill can improve drop test reliability as well [11, 12]. However the application of underfills increases both the cost of production and assembly cycle times in manufacturing and this must be considered against the reliability improvements. To reduce the costs of underfill application, corner bonding and edge bonding processes have been developed. In the corner bonding process, the adhesive is applied near the package corners before BGA packages or chip scale packages (CSPs) are placed onto wiring boards and solder reflowed. In edge bonding processes, the adhesive is applied to the edges of the BGA packages or CSPs after the solder reflow. The reliability of corner-bonded CSPs has been investigated [13, 14].

Failure detection systems and failure criteria used in the literature vary widely. There are three main failure detection methods used in drop test reliability: post-drop (static) resistance measurement [12, 15], event detection [16], and in-situ high-speed data acquisition [17]. The post-drop resistance measurement method measures resistance of solder joints after each drop. The event detection method determines if a failure event temporarily occurs during a single drop, but does not store the time-value history of the resistance during the impact. The in-situ high-speed data acquisition method measures the dynamic resistance of solder joints during and after the drop impact and board vibrations. Different researchers have used different failure criteria, for example, a resistance threshold of 300Ω [18], 1000Ω [1], or 1500Ω [16], or a resistance change of 10% [12], or 20% [15]. In a sense, all of these criteria are subjective, because, at this time, no scientific research has been done on the interconnection failure criteria. Determination of appropriate failure criteria is extremely important in order to observe first failures and when failures advance to different fail-

ure stages [17]. This variety of failure detection systems and failure criteria used by different researchers makes the comparison of results difficult, and as a consequence the industry still has many questions that need to be answered on the reliability of lead-free soldered microelectronics devices; the most appropriate solder alloys, board materials, and assembly processes for lead-free electronics are still being investigated.

1.3 Studying CSP Drop Impact Reliability

This paper presents the drop test reliability of 0.5mm pitch lead-free chip scale packages (CSPs). Drop impact test vehicles were assembled using a standard JEDEC drop reliability test board design [1]. Fifteen 0.5mm pitch CSPs were assembled on each of the 40 test vehicles with Sn3.0Ag0.5Cu lead-free solder. Eight test vehicles were edge-bonded with a UV-cured acrylic; eight test vehicles were edge-bonded with a thermal-cured epoxy; and twelve test vehicles were assembled without edge bonding. Half of the edge-bonded test vehicles were subjected to drop tests at a peak acceleration of 1500G with a pulse duration of 0.5ms, and the other half subjected to drop tests at a peak acceleration of 2900G with a pulse duration of 0.3ms. Half of the test vehicles without edge bonding were subjected to drop tests at a peak acceleration of 900G with a pulse duration of 0.7ms, and the other half subjected to drop tests at a peak acceleration of 1500G with a pulse duration of 0.5ms. The remaining six of the forty test vehicles were assembled without edge bonding and were subjected to drop tests for failure detection system validation.

Two drop test failure detection systems were used in this study to monitor the failure of solder joints: a high-speed resistance measurement system and a post-drop static resistance measurement system. The high-speed resistance measurement system, which has a scan frequency of 50KHz and a 16-bit signal width, is able to detect intermittent failures during the short drop impact duration. Statistics of

the number of drops to failure for the 15 component locations on each test board are reported. The effect of component position on drop test reliability is discussed. The test results show that the drop test performance of edge-bonded CSPs is five to eight times better than the CSPs without edge bonding. The drop test reliability improvement of edge-bonded CSPs with the thermal-cured epoxy is not significantly different from that with edge-bonded CSPs with the UV-cured acrylic; however the failure mode of the two edge-bond materials is different. The solder crack location and crack area are characterized with the dye penetrant method and optical microscopy.

It is known that the acceleration conditions endured by the test vehicle during the drop impact test are not identical at all component locations, and likewise that the strains in the test vehicle materials are not the same for all component locations during the test [1][22]. The component location dependent acceleration peak values were observed for several test vehicles, with and without edge-bonded CSPs, and with and without the data acquisition cable attached, and are reported for the 1500G drop impact condition. Analysis of these results indicate that the data acquisition cable attached to the test vehicle during the drop impact test influences both the symmetry of test vehicle response and the acceleration peak values at the component locations.

The following chapters contain a detailed description of this work. Chapter 2 describes the high-speed data acquisition system developed at Cal Poly to enable this study. Chapter 3 covers the drop impact reliability test vehicle design and the procedures that were followed to conduct the tests. Chapter 4 discusses the reliability data analysis, failure analysis, and acceleration conditions observed on the test vehicle. Chapter 5 draws conclusions about the results and summarizes the contributions of this thesis. Supporting materials are contained in the Appendices.

Chapter 2

Design of a Data Acquisition System

2.1 Requirements

The primary goal of a high-speed data acquisition system for drop impact testing is to capture accurate data indicating the condition of solder interconnections during the test. Drop impact tests are a short duration test (less than one second) that is repeated many times. This presents two problems for the test system: 1) the duration of the test being very short requires that the test system be capable of sampling accurate data at a high frequency, and 2) the test is repetitious and the time necessary to conduct the testing must be minimized, therefore each cycle must be finished as quickly as possible. However, these goals are in conflict because high frequency sampling, with high accuracy data, can result in large and time consuming data storage tasks.

The JEDEC drop test standard [1] specifies a variety of drop impact conditions that may be used. The drop impact conditions are described by a half-sine acceler-

ation pulse which has a specific peak and duration. Figure 2.1 is an example of the general acceleration pulse. Each acceleration pulse has a different duration during which the acceleration force is applied to the test vehicle. The longest commonly used pulse duration is 1.0 ms (one-thousandth of a second) and the shortest is 0.3 ms. The objective of the test system is to observe the condition of the solder interconnections as the test progresses rather than to only determine if failure occurred afterward. Therefore, the test system needs to obtain a series of measurements within the acceleration pulse duration, and continue to collect measurements for some length of time afterward. The JEDEC standard suggests a minimum sampling rate of 50Khz, which corresponds to 50,000 samples per second, or 50 samples taken during a pulse duration of 1 ms.

For our test system the target sampling rate of 50Khz was selected, which is adequate to obtain several samples during the initial shock pulse of the drop impact. Figure 2.2 shows an example of the samples that may be obtained at a 50Khz sampling rate during a typical 1500G-0.5ms acceleration pulse. The primary deflection time of the board and first harmonic vibration frequency in a 1500G drop impact test are near 4 ms and 240Hz [20] respectively; with a 50Khz sampling frequency this system provides more than 200 samples per board deflection cycle. As the board deflects broken solder interconnects may be stretched apart and temporary discontinuity could be detected in the connection, therefore it is important to continue measuring continuity during at least the first few board deflections that occur after the impact. Some researchers have suggested that prolonged vibration is a contributing factor to failure, so sampling for a longer period after the impact may also be important and the test system should be configurable for a period of sampling time after impact.

The test system must also be designed to optimize the testing process for many test repetitions. A typical drop test vehicle may be dropped up to 300 or more times before the test completes. To control the data acquisition system a software program

is needed which will aid the user in starting the test, obtaining the data, saving the data for later analysis, and preparing to run the test again.

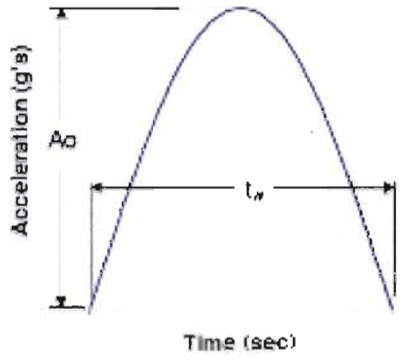

Figure 2.1: General half-sine input acceleration pulse

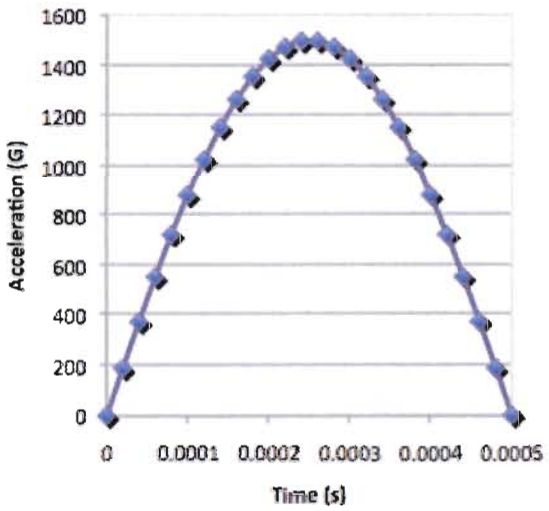

Figure 2.2: Example acceleration pulse with 50Khz frequency sampling

2.2 Design and Assembly

The high-speed data acquisition system should be built from a cost-effective desktop computer platform. An analog-to-digital converter (ADC) would be needed to measure the resistance of the solder interconnections, and various hardware would be

needed for connectors, cables, and power supply. A Windows-based software control program written in C++ was developed to interface to the ADC control libraries provided from National Instruments for the chosen ADC hardware. The resistance measurement is realized by using a voltage divider circuit and sampling the voltage at the divider nodes using the ADC.

2.2.1 Resistance Testing Circuit

The solder interconnects being tested are arranged as a ball-grid array (BGA) underneath a CSP. All the solder interconnects under a single CSP are connected in a daisy-chain such that the array has a single input and a single output, with all the other connections wired together forming a single long wire. The connections for the daisy-chain are made both inside the test package and in copper trace routing on the test board. The resistance of this daisy chain is the measurement of interest in the study. As the solder interconnections slowly fail during the test, or if any single solder interconnect fails completely, the resistance of the daisy-chain should change [21].

A simple and proven method of achieving dynamic daisy-chain resistance measurement at near real-time was used [21]. The component daisy chain is placed in a DC series circuit with a static resistor (R_{Static}) of known value, in this case 100Ω, to construct a voltage divider circuit as shown in Figure 2.3. The DAQ records the voltage (V_{Comp}), divided across the component resistance and static resistance. The voltage (V_{Comp}) relates to the resistance (R_{Comp}) by Eq. 2.1, where V_{DC} is the DC voltage source, in our system set to 5V. As the component electrically fails, the resistance rises ($R_{Comp} \Rightarrow \infty$) and the DAQ registers a rise in voltage (i.e. $V_{Comp} \Rightarrow V_{DC} = 5V$).

$$R_{Comp} = \frac{V_{Comp} \cdot R_{Static}}{V_{DC} - V_{Comp}} \tag{2.1}$$

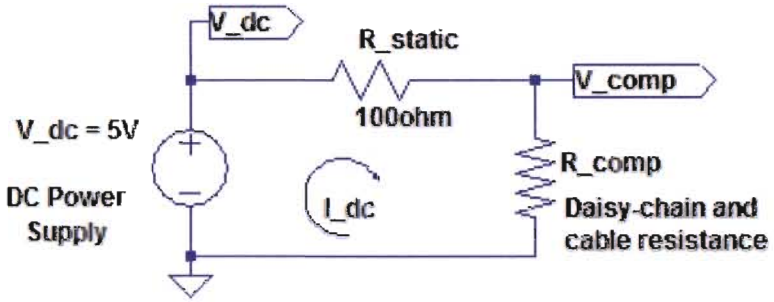

Figure 2.3: DC series voltage divider circuit

2.2.2 Data Acquisition Hardware

The ADC that was selected is a National Instruments DAQ M-series PCI card, model #6254, that supports analog inputs within the targeted 0-10V range and produces digital results at 16-bit accuracy. The card is connected to a standard desktop computer with a Pentium IV processor and 80Gb storage. The PCI bus from the computer provides a stable, regulated 5V power supply; this was chosen as the supply to the voltage divider circuit to eliminate the need for an external power supply device. The card supports up to 32 input channels, and has a maximum sampling frequency of 1Mhz which is divided between all the active channels. For our application 17 channels would be needed, 15 for the CSPs being tested, and one each for the power supply voltage and trigger signals. Supply voltage is measured during the test to verify that the power supply signal does not vary, which could result in different voltage divider levels and could be interpreted incorrectly as a change in resistance of the solder interconnections. The trigger signal is used by the software program to begin collecting data just prior to the impact of the drop table and machine base in order to minimize the amount of data storage for each test repetition.

A trigger was added to the system by using an Infrared optical sensor, which produces a high voltage pulse while the sensor is blocked. The sensor was clamped to the test machine to hold it in place where a knife-blade would pass through the sensor

notch just before the table impact, which triggers the system to begin recording data. With a DAQ system collecting data at such a high frequency it is important to limit the collected data to the smallest useful time period, otherwise the total data volume would become unmanageable quickly. The system software must control when to stop recording; this is done by configuring the software to limit the total number of samples taken.

The complete system uses a National Instruments (NI) ADC, a desktop computer, and a voltage divider network contained in two NI connector boxes, model #SCB-68, shown in Figures 2.5 and 2.6. The cables attach to the connector boxes with a custom made spliced cable (Figure 2.7), and the connector boxes are attached to the ADC card via NI parallel cables. All the cable connections are secured during testing with retention screws. The front panel of the boxes were replaced by machined sheet metal brackets so that a standard 32-pin parallel cable connector could be used in each box front.

A low voltage range is preferred for the supply to reduce the power used in the test and to prevent arcing as the solder interconnects fail. The static resistance used in the circuit is chosen to match the suggested failure condition for the DAQ system [1]. That static resistor choice sets the failure condition at half the supply voltage, or when the resistance in the daisy chain matches the static resistance. Also because the static resistor is small, if a high voltage supply is used then the current in the daisy-chain would be excessively high and the wiring board is not designed for high current.

The NI connector boxes include a fused supply circuit from the PCI bus with a 0.75 amp fuse. Each of the 15 channels of the system would be running a 5 volt supply through a voltage divider network that will have a minimum of 100Ω resistance, so if all 15 channels were assembled in one connector box the 0.75 amp fuse would be

overloaded. The conservative decision was to not change that fuse to a higher current limit, because the ADC card would then be stressed and it was the most expensive and critical part of the system. The channels were split into two groups, with the first 8 channels in one connector box and the remaining 7 channels in the other, so that each connector box would have at most a 0.4A current draw ($8 \times 5volts/100\Omega = 0.4amp$). The first box also has the trigger input, and the second box also has the supply voltage input, neither of which draw any current from the ADC supply voltage.

The first prototype system used a separate resistor network circuit board, on which the static resistors and voltage divider network was built (shown in Figure 2.4). This system required an external power source and had inconvenient connections to the cables. The voltage divider circuit was later assembled directly inside the connector boxes which have bread-board style through-holes arrayed in the center of the boards between the input connector rails. The resistors and wires were inserted through the board and solder bridged on the bottom side to create the network. Solid-core 24-gauge copper wire was used inside the connector boxes. The resistors used were selected for the closest possible match to 100Ω using 5% accuracy resistors. The selected resistors for each channel had resistances shown in Table 2.1; both the ADC channel number and the corresponding test vehicle component location are indicated.

Table 2.1: Static resistor assignments for voltage divider circuit

Connector Box 1			Connector Box 2		
Component	**Channel**	**Resistance (Ω)**	**Component**	**Channel**	**Resistance (Ω)**
R1	**AI1**	**100.0**	**R9**	**AI17**	**98.9**
R2	**AI2**	**99.9**	**R10**	**AI18**	**99.3**
R3	**AI3**	**99.5**	**R11**	**AI19**	**98.9**
R4	**AI4**	**99.8**	**R12**	**AI20**	**99.0**
R5	**AI5**	**99.8**	**R13**	**AI21**	**99.0**
R6	**AI6**	**100.5**	**R14**	**AI22**	**99.8**
R7	**AI7**	**99.9**	**R15**	**AI23**	**99.2**
R8	**AI8**	**99.6**			

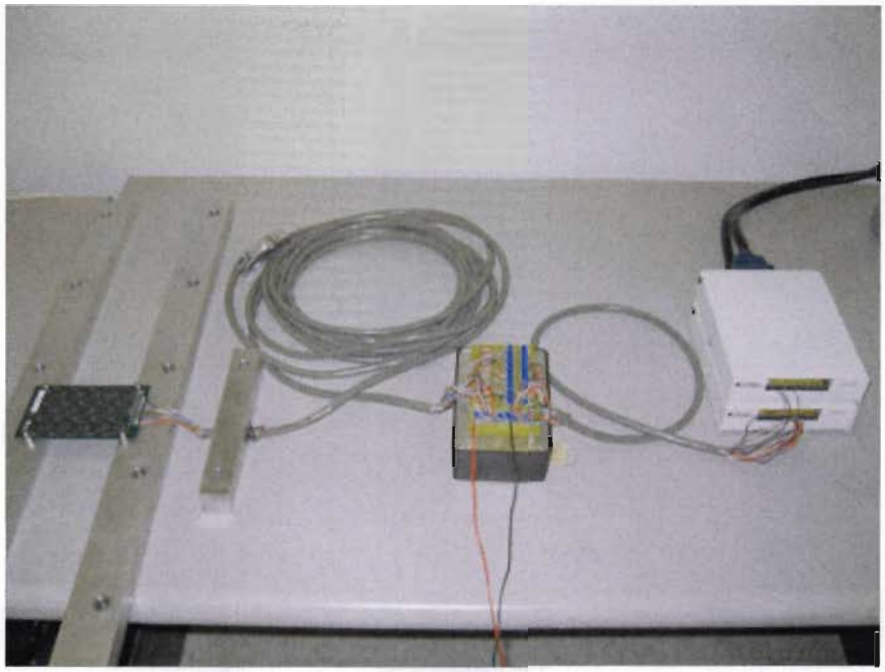

Figure 2.4: NI connector box, cables, junction circuit board, and test vehicle mount.

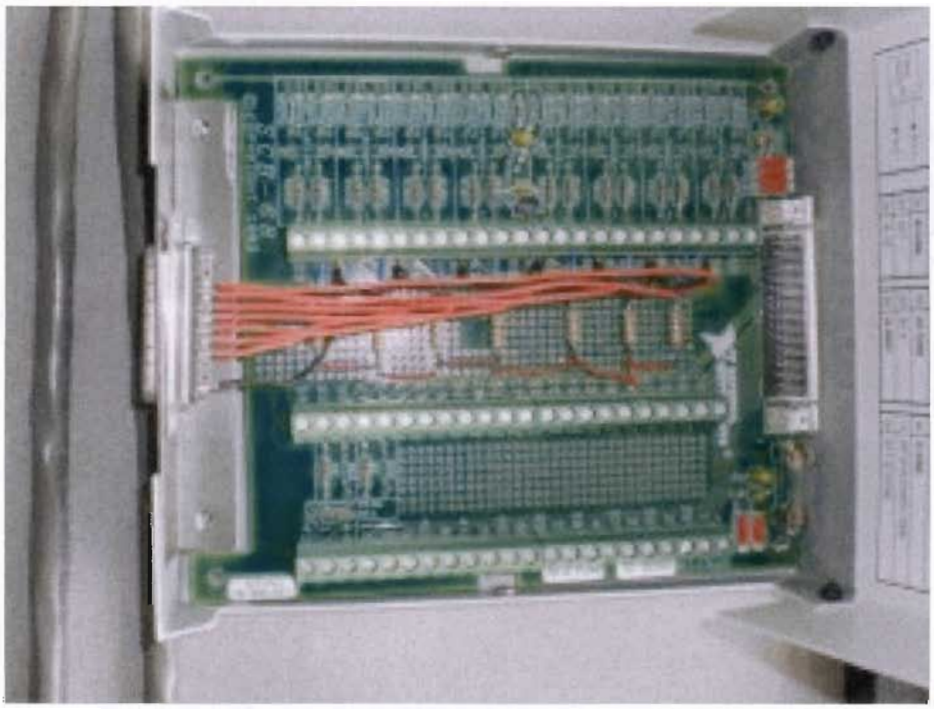

Figure 2.5: Voltage divider circuit assembled inside NI connector box

Figure 2.6: NI connector box from front

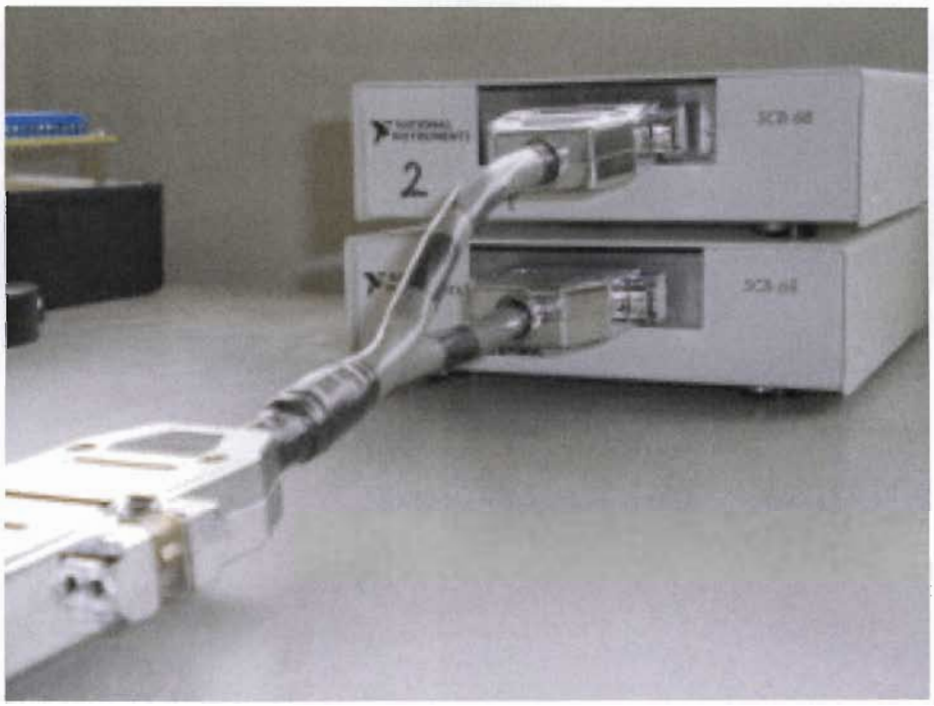

Figure 2.7: NI Connector boxes with spliced cable connected

2.2.3 DropGather: DAQ System Control Software

A console program named `DropGather` was written in C++ as the user interface and control software for the data acquisition system. The program is written to run on Microsoft Windows XP but may run on other operating systems with a few changes provided the libraries it uses are available for those systems. The program links to the National Instruments `DAQmx` libraries supplied for use with the ADC, and provides a software interface to the control routines in the library, as well as providing opportunity for the test system operator to configure test parameters to identify the data files that are produced. The data recorded by the software is output in multiple formats, as tab-delimited text for statistical software, and as plotted graphics in PDF format.

The `DropGather` software has been released as open-source software under the terms of the GNU General Purpose License version 3 and may be used, copied, or modified by others under the terms specified by that license. More information on the licensing may be found at the Free Software Foundation website, www.fsf.org.

Software Design

The software needs to handle the following tasks: 1) obtain configuration input from the user for the current test parameters,2) initialize the ADC hardware and open driver resources to the device, 3) configure the hardware for the drop test system requirements, 4) react to the trigger signal and capture voltage data for each channel, 5) save the data for future analysis, 6) display rapid feedback information on the test results to the operator.

The software is written using classes in C++ but is essentially a sequential program. It runs through configuration, begins the testing cycle, conducts a single test,

then if the operator wants to run another test the program loops to begin a new testing cycle. To configure the software for a new test vehicle the program should be closed and restarted.

The main function is defined in the `dropgather.cpp` file; the configuration input from the operator is handled here. There are four C++ classes used in the program: `imeProject`, `DataCollection`, `TaskHandler`, and `Excpt`. The `imeProject` class configures the hardware, runs the task, and processes the data into output files and on-screen plots; this is the primary worker class for the program. The `DataCollection` class is a wrapper class for accessing the data in the memory of the ADC. The `TaskHandler` class is a wrapper class for the control functions that define a task in the `DAQmx` library, and allows improved error handling around those functions. The `Excpt` class is a special exception defined to workaround a quirk in the string handling of the `DAQmx` library error messages.

Test Parameters. The program starts with the user selecting whether the current test vehicle will be considered 1-Fail (that is, removed from the testing process after the first component fails), has lead-free or lead-based solder, OSP or ENIG coating, and whether any underfill was applied. Then the test vehicle number is entered, and the operator is asked to verify the selections. The text console and user prompts are shown in Figure 2.8. The software uses these configuration parameters in the file name for the data files that are generated by each test run until the program is closed.

These choices are not repeated until the program is quit and restarted. Once the test parameters are input the program allows the operator to do multiple test trial runs, which are assumed to be on the same test vehicle. The current test run number is saved between trial runs; it is also saved if the program is quit.

When the operator changes to a new test vehicle the current test run number must be manually reset by editing the index.ini file and setting the run number to

Figure 2.8: DropGather software console output

zero. Setting the run number to zero allows a single pretest trial run to be conducted to provide initial condition data for the test vehicle.

Initialization and Hardware Configuration. The `DAQmx` library has a primary header file for C++, `NIDAQmx.h`, which provides function declarations for acquiring access to the hardware device. The general process for reading data from the device is to create a task to be completed by the ADC, start that task, wait for the task to complete, then retrieve the data from the ADC.

The initial hardware configuration is handled by the `imeProject` class constructor which creates the memory regions for the data to be locally stored, then configures the data channels to be read and the length of the task to be completed. The channels to be read are configured using the `DAQmxCreateAIVoltageChan` function by setting the string that defines the channel names to use. This function also sets the minimum and maximum voltage for the analog input. The length of the task is set by the `DAQmxCfgSampClkTiming` function call; the desired sampling rate is set to 50000 per second, and the total number of samples is set to determine how long the task will

run.

The ADC input channels used are `AI0` (trigger), `AI1-AI8` (R1-R8), `AI17-AI23` (R9-R15), and `AI24` (supply voltage). The channel definition was chosen to keep all 15 component voltage channels contiguous in memory to simplify the data file output routine.

Data Collection. The task configured for the ADC is a blocking operation which waits until the trigger signal occurs before beginning the data collection. When the trigger signal occurs, caused by the drop table falling passed the optical switch, the ADC begins collecting the samples from each channel until the end of the task. The expected trigger signal is a rising-edge square-wave voltage signal.

The data that is read by the device is stored internally in its onboard memory, so there is a limitation to the total number of samples that can be stored in a single task, but the limit is much larger than necessary for drop tests of 1-2 seconds total sampling time. The data is collected in a contiguous memory region as a multidimensional array that must be copied from the device memory to the main system memory before being output in files or graphs. Since that memory copy is a slow and resource expensive process it can only be done after the task completes.

Output of Test Results

Data Storage. The data must be stored to disk in a format that is easily readable by a computer program for later analysis. For this purpose the tab-delimited text format was chosen which produces a plain text file containing the decimal equivalent sampled voltage values that were recorded for each channel. At the beginning of the file a column header is printed, then all the data rows follow with a single tab character separating each value in the row. The sampled voltage values are not converted to resistances by this version of the software.

This file format can be easily loaded into spreadsheet or statistical software packages. The tradeoff for this compatibility is a much larger file on disk than a binary format would require, but has the additional benefit of being human readable if necessary.

Failure Plots. During the test sequence the test operator/user must verify that no abnormal conditions occur such as a broken cable connection to the test vehicle. To make this task easier, rapid visual feedback is provided on the test results for each sampling task, or trial run. The collected data is plotted to a window on the computer as a scatter plot of voltage versus time. Each sampling channel is plotted separately, one per window, and the user must examine each window then close it before the test system can be reset for the next trial run. To plot the data the *DISLIN* scientific data plotting library is used.

In addition to the on-screen plots that are shown to the operator, a PDF format file is written to disk with each plot on a separate page which can be used to quickly review the test results from prior runs. The PDF file was also used to verify the manually recorded failure event charts after the testing was completed.

The data file written to disk is typically 7.2MB per test run, for a sampling period of 1 second; this generates a text file with 50,000 lines, and each line has 17 tab separated decimal values. The PDF file written to disk is typically 1.4MB per test run and produces a 17 page document. The desktop computer system had two 40Gb hard disks; one of these had Windows XP and supporting software installed on it. The total storage space available was about 60Gb and would be more than adequate, because the project drop testing generated 24Gb of data files.

2.3 System Operational Testing

2.3.1 DAQ System Capabilities

The high-speed data acquisition system was tested at Cal Poly with a MTS 886 shock test machine (shown in Figure 2.9). Using this machine the maximum achievable acceleration peak value was 680G with a nominal pulse duration of 2.0ms. This machine has a very large and heavy drop table which can be raised up to a drop height of almost 60in. However because of the extreme weight of the drop table the machine base is mounted on large rubber cushions which prevent damage to flooring underneath the table feet. The cushions also cause a much broader shock pulse with lower peak acceleration value than a machine without such cushions. The peak acceleration value and pulse duration for this machine had a greater than 10% variation, most likely due to the old rubber cushions, which was an undesirable source of error.

Since the JEDEC standard [1] drop test conditions were not met by this machinery an alternative drop testing machine was needed to conduct the drop impact reliability study. For the drop impact reliability study a Lansmont M23 TTS II Shock Tester was used, courtesy of Henkel Corporation. However, the initial operational tests of the data acquisition system were conducted on the MTS 886 machine at Cal Poly.

The capability of the DAQ system to detect failed solder interconnections was tested with assembled test vehicles. A test vehicle was attached to the DAQ system, and then repeated drop tests were conducted. The failures of the components were recorded and each was verified with manual resistance probing to determine whether the data acquisition system correctly identified loss of continuity. This testing identified several issues with the test system which needed to be corrected.

1. The initial software design plotted more than one component's voltage data at once in a single window using different colors. With these plots it was very

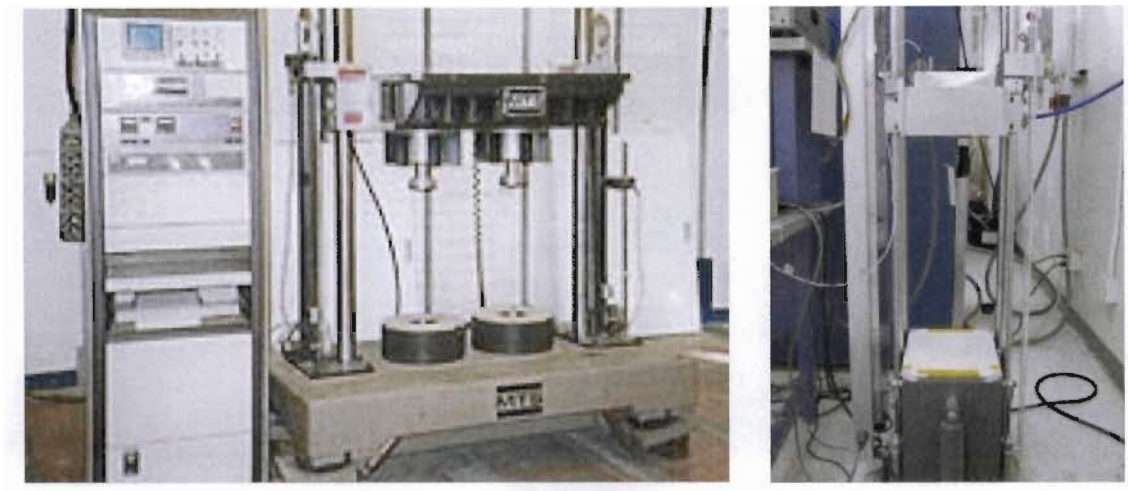

Figure 2.9: MTS 886 Shock Tester (left) and Lansmont M23 Shock Tester (right)

difficult to determine which components had failed on the test vehicle for each trial run. To correct this problem the plots were separated out to individual windows to speed up the process of review during each trial run. Additionally, the color used to plot the data on-screen now alternates for each plot so that the operator can easily realize if a window was closed accidentally without reviewing the plot. Both of these changes significantly reduced the time needed to review the results during the testing process.

2. The software requested operator comments to be entered after each drop test so that these comments could be saved in the tab-separated text file along with the recorded data. The expected comments would be notes about which components failed, whether any abnormal conditions occurred such as a broken wire connection to the test vehicle, or other notes the operator may need to record. However, since the program takes a significant amount of time to write the data file to disk this waiting period while the operator entered comments was inefficient for the program workflow. Since a second computer was available during the drop testing the comment feature was removed and handled through

a spreadsheet; the data file could then be written to disk much faster while the drop test machine raised the table for the next drop.

3. The length of data recording was originally planned for 2 seconds, but after initial testing this was reduced to one second which continues sampling until the test vehicle has nearly stopped vibrating. Sampling for the full 2 seconds would have almost doubled the data volume without a clear purpose.

During operational testing of the DAQ system a detailed set of procedures were developed for use with the MTS 886 shock test machine. These were later revised (see Appendix A) to improve process efficiency for use with the Lansmont M23 shock test machine and to reflect changes in the software for process optimization.

2.3.2 Drop Test Calibration

To achieve a desired drop impact loading condition the drop test machine must be calibrated by repeated observations and adjustments. The loading condition is typically described by an acceleration pulse peak A_0 and duration t_w (refer to Figure 2.1).

The following general equations describe the drop impact acceleration pulse and the controllable parameters in the test. Equation 2.2 describes acceleration as a function of time, where A_0 is the maximum acceleration peak value and t_w is the half-sine pulse duration. The drop height H and rebound coefficient C are the controllable parameters and, as Equation 2.3 describes, determine the value of A_0 and t_w.

$$A(t) = A_0 sin(\frac{\pi \cdot t}{t_w}) \tag{2.2}$$

$$\sqrt{2gH} = \frac{2A_0}{C}\frac{t_w}{\pi} \tag{2.3}$$

To calibrate the test parameters, drop height and impact surface, an accelerometer and signal conditioner can be used to accurately observe the loading conditions of the drop impact. Calibration for this project was done using the following test equipment:

1. A PCB Piezotronics model # 352A25 accelerometer, which is a lightweight (0.6 gram), miniature device. It is factory calibrated to output 2.5 mV/G signal and respond accurately at up to 2000G.

2. Lansmont Test Partner TP-3 signal conditioner and software. The signal conditioner samples and interprets the analog voltage signal from the piezoelectric accelerometer and displays the results, as peak acceleration value and duration of that pulse, and graphically as a function of time. The signal conditioner must be configured for sensitivity, matched to the accelerometer being used, and desired signal filter frequency.

The drop height is easily changed by the drop test machine which has digital input controls for the height the table is raised to before the drop. However, the rebound coefficient must be controlled by varying the impact surface between the raised table and the machine base. For the MTS886 machine this was controlled by felt pads and optional gas pressurized cushions; however because of the rubber cushions underneath the machine base already made the impact too gentle, neither the felt or gas pressure were used on this machine; the acceleration pulse was calibrated using only drop height for this machine. For the Lansmont M23 shock test machine, the acceleration pulse was calibrated using both drop height and impact surface. The impact surface used was dense sheets of paper or felt, with the number of sheets dependent on the acceleration pulse width desired.

A 5 Khz low-pass filter was applied to the analog accelerometer signal by the Test Partner software. The filter cutoff frequency is configurable and several values in the range of 500 Hz – 10 Khz were tested before the 5 Khz frequency was chosen. The filter helps to interpret the output of the piezoelectric accelerometer as a smooth curve of the acceleration condition over time. A filter cutoff frequency that is too low will reduce the observed acceleration peak value and rate of change at any time. A filter cutoff frequency that is too high will allow signal noise to be included in the graph and may show impossibly high rate of acceleration change. An appropriate filter frequency must be chosen based on reasonable estimation for the test conditions being used. For the accelerometer and drop conditions being used in this study a filter frequency of 10 Khz resulted in a significantly lower peak G-value, but a frequency of 5 Khz did not reduce the peak G-value significantly. This filter is used only for the accelerometer output and no filter is used for the data acquisition channels which sample the component resistance during the test.

2.3.3 System Modifications for Lansmont Drop Tester

Due to the change from the MTS 886 shock test machine to the Lansmont M23 shock test machine for the drop impact reliability study, it was necessary to make some changes to the data acquisition system to integrate it with the new shock test machine.

DAQ system cable clamps

The data acquisition system prototype design included a test vehicle mount and cable clamping system (shown in Figure 2.4) that was fitted to the MTS 886 machine drop table. The cable was routed underneath an aluminum bar which was bolted down to the table next to the test vehicle mount. The cable restraint prevented the

weight of the cable from pulling against the through-hole solder joints in the test vehicle and possibly causing false-positive failures. For the data acquisition cable to be clamped to the Lansmont M23 drop table a new clamp was required. As shown in Figure 2.10 two 1/2" cable clamps were bolted into the side of the drop test table. The drop table top-plate was removed from the machine to drill and tap for the two #10-24 threaded machine screws to mount the clamps. The clamps held the cable in position without pulling against the solder joints or letting the cable get trapped underneath the dropping table.

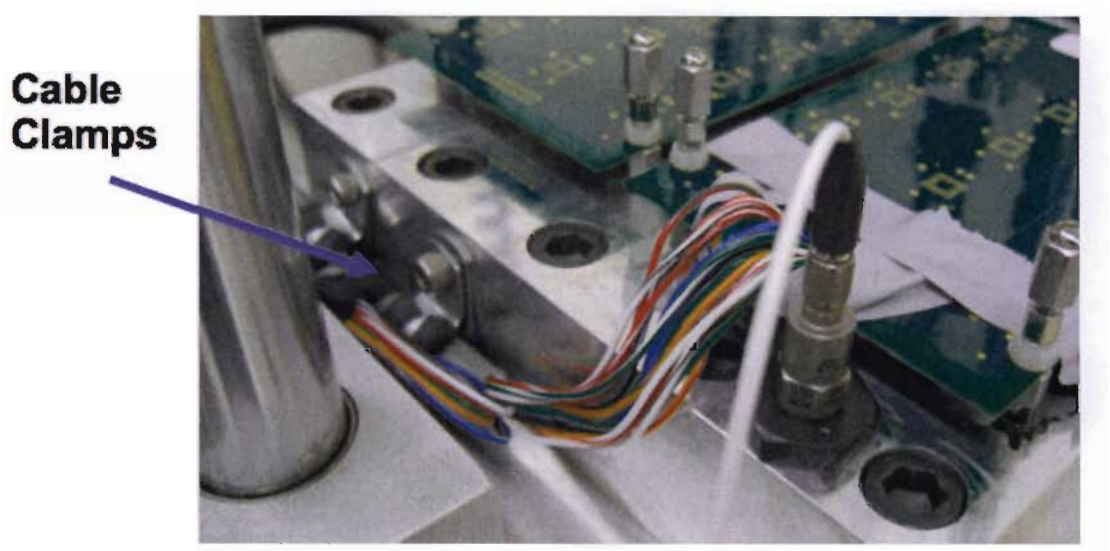

Figure 2.10: DAQ system cable clamps on Lansmont M23 drop table

DAQ data collection trigger

The original trigger design used an optical sensor as discussed in Section 2.2.2. The Lansmont M23 machine had no convenient location for mounting the optical sensor. The optical sensor wiring was disconnected, and an alternative triggering system was easily adapted to work with the system. The electronic control box for the Lansmont machine has a configurable external trigger signal which will automatically issue a trigger pulse when the drop table reaches a specific height above the machine base.

The Lansmont external trigger produces is a single square pulse with pulse width of 500 milliseconds and amplitude of 24 volts. The DAQ system expects a square pulse with an amplitude between 0 and 10 volts. The DAQ system is configured to initiate the data collection task when a rising-edge pulse is detected on the analog input AI0 which is connected to the trigger signal.

To adapt the external trigger signal to the DAQ system a voltage divider network of three 1kΩ resistors was used as an interfacing circuit to split the 24 volt signal down into 8 volt segments (Figure 2.11). The 8 volt signal level in this circuit was connected to the DAQ trigger input. This circuit was added inside the NI connector box in which the trigger input is connected to the ADC. The trigger height, which is the distance the table has left to fall before impact with the machine base, was set to 1.5 inches for all conducted tests.

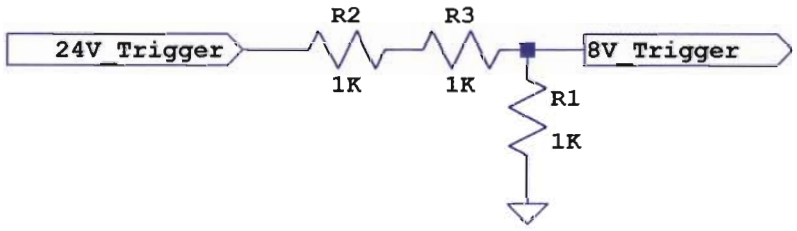

Figure 2.11: DAQ trigger voltage divider interfacing circuit

Chapter 3

Drop Impact Reliability of CSPs

3.1 CSP Reliability Test Vehicle

A drop impact reliability study requires a test vehicle which is representative of the electronic device being tested. The test vehicle is an electronic assembly, made up of a printed wiring board (PWB), also called a printed circuit board (PCB), and electronic components. The components are soldered onto the test vehicle as they would be used in an electronic device. The solder interconnections formed between the components and the test vehicle are the mechanical and electrical connections being tested in drop impact tests.

3.1.1 Test Vehicle Design and Assembly

The test vehicle was designed according to the JEDEC standard [1]. It uses an eight-layer FR4 material board (layered fiberglass) with board dimensions of 132mm by 77mm and a thickness of 1 mm. The components used were 0.5mm pitch Amkor CSPs having 228 I/O and with a size of 12mm by 12mm. The CSP has daisy-chained connections with an input and output trace located at one package corner. The

boards have Organic Solderability Preservatives (OSP) surface finish on non-solder mask defined (NSMD) pads, while the components have electro-plated nickel-gold surface finish on solder mask defined (SMD) pads. The test vehicle with components assembled is shown in Figure 3.1. Figure 3.2 shows the perimeter ball-grid array configuration of the CSP solder balls as well as relative size of the CSP next to a penny. Figure 3.3 illustrates how the daisy-chain connections are routed between the component and PWB.

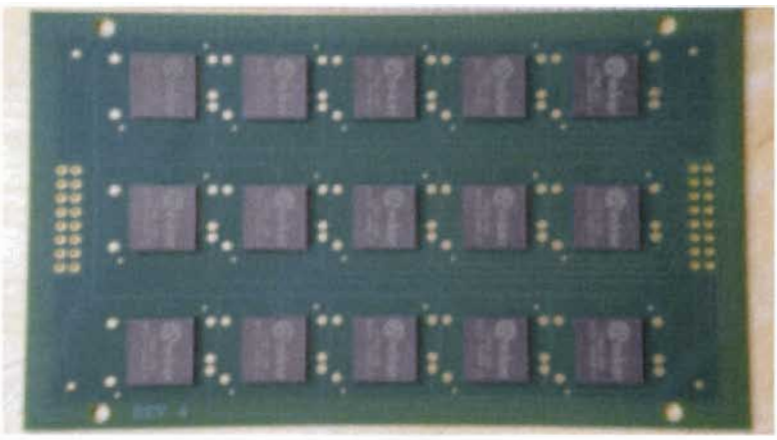

Figure 3.1: Test vehicle with components

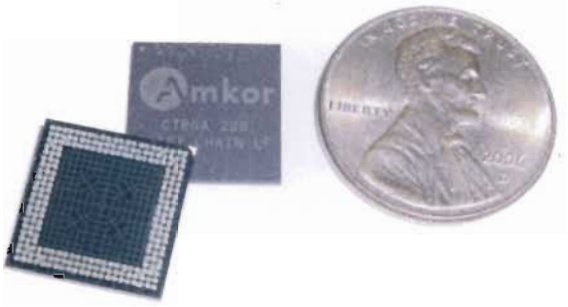

Figure 3.2: Amkor 12mm perimeter ball-grid array CSPs (top and bottom)

Sn3.0Ag0.5Cu (SAC305) Multicore 318LF lead-free solder paste (Type 3) was stencil printed using a DEK machine through a 4 mil thick electro-polished stencil with 12 mil square apertures. Solder paste height and volume were measured by a CyberOptic machine to ensure that a high quality stencil print had deposited the

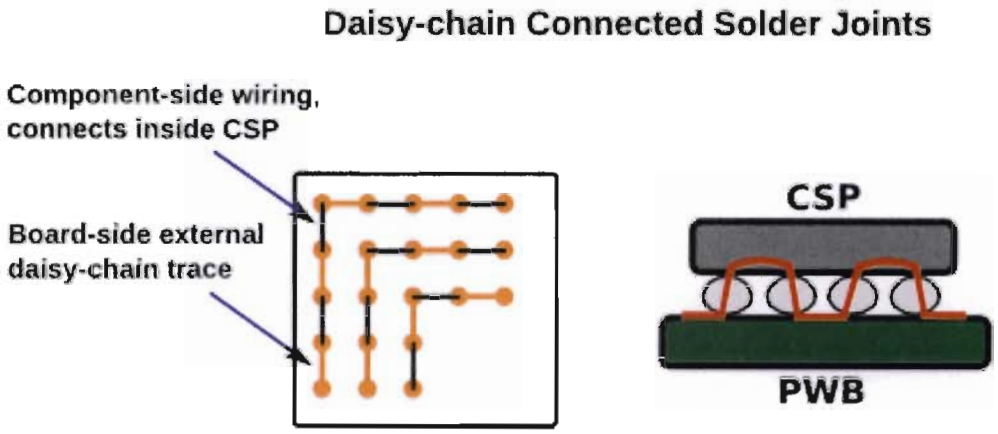

Figure 3.3: Illustration of daisy-chained solder joints in the CSPs

correct amount of solder paste on each board. The component was picked and placed by a Siemens F5 machine. A Heller EXL1800 oven with seven heating zones and one cooling zone was used for solder reflow. The reflow oven processing was done in air. The reflow profile is shown in Figure 3.4.

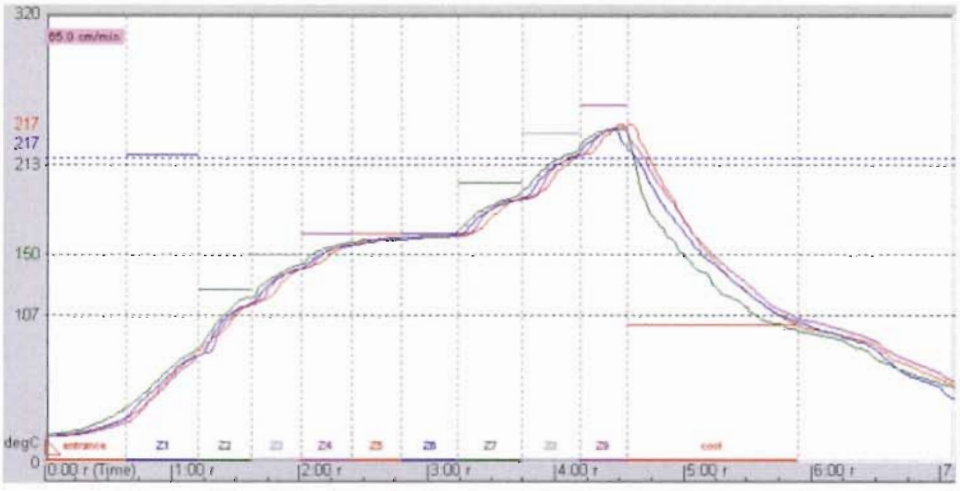

Figure 3.4: Solder reflow profile

Post-assembly cross-sectioning and SEM imaging showed good solder joints with some small voids as shown in Figure 3.5. Some irregularities in solder ball shape were found but no fractures were identified before drop testing. Inspection by optical microscopy and X-ray imaging, as shown in Figure 3.6, showed shiny, round and

well collapsed solder joints with no bridging. These indications confirmed that the assembly process was well suited to the solder paste and surface finish used for these test vehicles, and had therefore resulted in a high quality assembly process for these drop impact test vehicles.

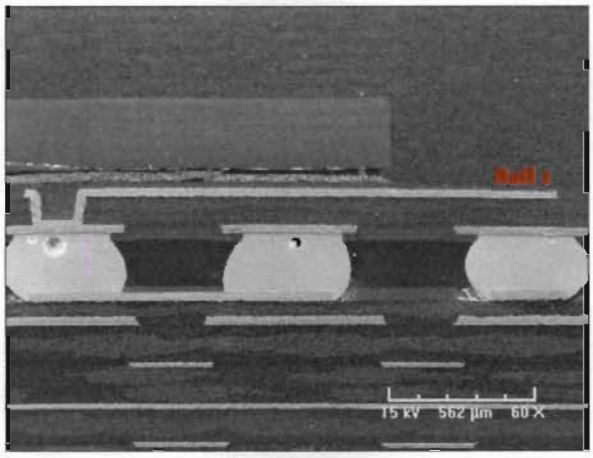

Figure 3.5: SEM image of cross-sectioned solder joints after assembly

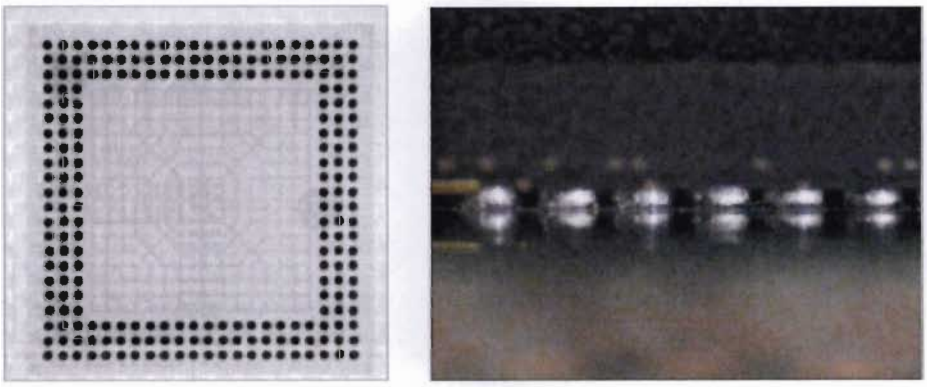

Figure 3.6: X-ray (left) and Microscope (right) images of CSP solder joints

3.1.2 Edge-bonded CSPs

Edge-bonding electronic components is a post-assembly process for strengthening the mechanical attachment of the component to the circuit board. A glue material is dispensed a short distance along the corners of the component package so that the glue

will adhere to both the package sides and to the circuit board. The bond formed by this glue material is intended to share some of the stresses on the solder interconnects near the package corners while the circuit board bends under mechanical shock. Edge-bonding is known to improve reliability for larger components, but is typically not used for CSPs. The expectation of this study is that edge-bonding will significantly improve drop impact reliability of CSPs and prove to be a useful process for this class of components. Capillary underfill processes have been more commonly used for CSPs, however edge-bonding is a much faster and cheaper process. Therefore, if adequate drop impact reliability improvement can be obtained by edge-bonding CSPs this would be a valuable finding for the electronics industry.

The test boards were divided into three cells, one of which was edge-bonded with a thermal-cured epoxy, one of which was edge-bonded with a UV-cured acrylic, and the third cell having no edge-bonding. The edge-bond was applied on all four package corners by an Asymtek Century series machine. The UV-cured acrylic material was cured per manufacturer recommendations by exposure to a UV lamp, emitting 365nm wavelength light, for 80 seconds. The thermal-cured epoxy material was cured per manufacturer recommendation in an air atmosphere oven at 80 degrees Celcius for 20 minutes. The edge-bonds had an average length of 3.81 mm (150 mils) along each side (measured in both directions away from each corner) and an average length of 1.2 mm (47 mils) fillet leg after being cured.

Figures 3.7-3.8 show examples of the finished edge-bonds for each edge-bond material. In Figure 3.8 the edge bonds on each package are shown to be adhered to the sides of each package as expected. The finished edge-bonds were visually inspected for major defects before drop testing and none were found.

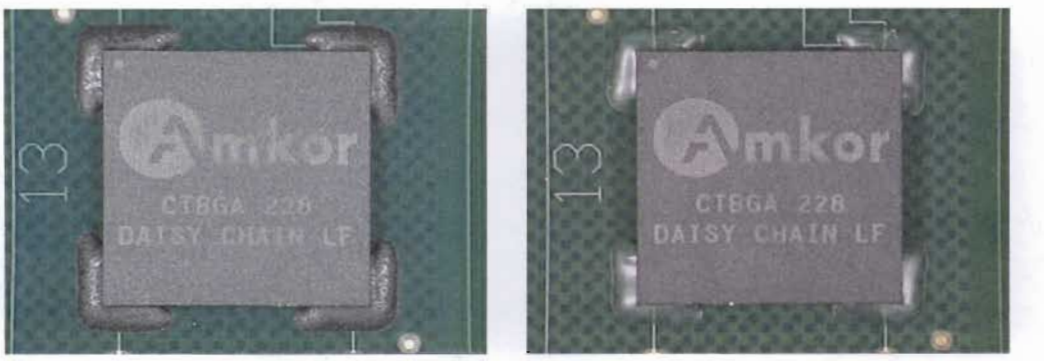

Figure 3.7: Edge-bonded CSPs Top View, Epoxy (left) and Acrylic (right)

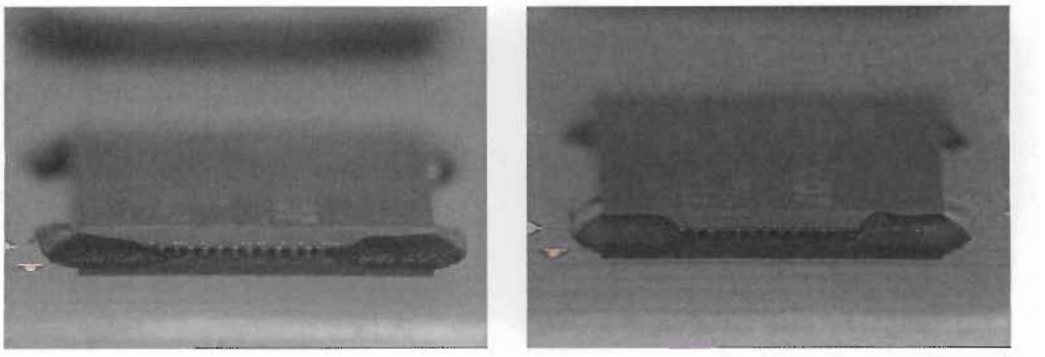

Figure 3.8: Edge-bonded CSPs Side View, Epoxy (left) and Acrylic (right)

3.2 Drop Test Methodology

The drop tests were conducted using a Lansmont M23 TTSII shock test system, which applies a single half-sine shock impact pulse to the test vehicle for each drop. Multiple impacts were prevented by a pneumatic rebound brake system that slowly brings the table to a stop. The test vehicle is permitted to vibrate after the drop impact but will not be allowed to bounce against the impact surface more than once per drop. This shock test system represents a typical board level drop test setup.

For this study the test vehicle was mounted with the board in a horizontal position with the components facing downward, as in Figure 3.9, which is the most severe orientation for board deflection [1, 19]. Four corner shoulder screws with 12 mm standoff supported the board mounted on the drop table. The drop table was secured between two guiding rods and could travel only along the vertical direction. When dropped from the chosen height, the drop table falls vertically and impacts the stationary seismic shock mounted table base. This impact transfers an input acceleration pulse to the test vehicle through the four PWB corner support screws. The acceleration peak value and pulse duration are influenced by the drop height, friction against guiding rods, and impact surface; the drop height and impact surface are the controllable factors.

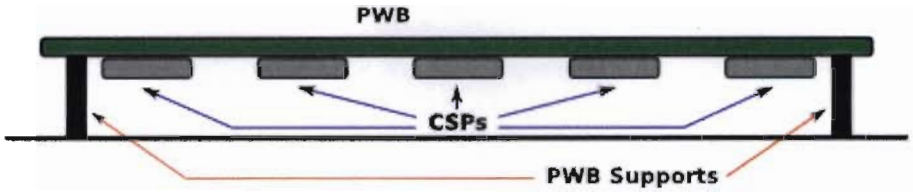

Figure 3.9: Test vehicle orientation when supported on drop test table

Three acceleration conditions were chosen from the JEDEC recommendations [2]: 900G, 1500G, and 2900G, with 0.7 ms, 0.5 ms, and 0.3 ms durations respectively. These are JEDEC condition F, B, and H. The input acceleration pulse was monitored

during each drop by two PCB Piezotronics accelerometers attached to the table base plate. To set the acceleration condition, the drop height and impact surface were selected by using the average result of both accelerometers to calibrate the system. One of the acceleration pulses is shown in Figure 3.10. The table impact surface varied between acceleration conditions; a felt pad and one sheet of paper was used for 900G and several sheets of watercolor paper were used for 1500G and 2900G. The drop heights used were 36.8 cm (14.5 inch), 57.2 cm (22.5 inch), and 76.2 cm (30 inch). The input acceleration condition on the drop test table was measured for every drop, and the drop height was adjusted incrementally to maintain consistent acceleration conditions during test cycles.

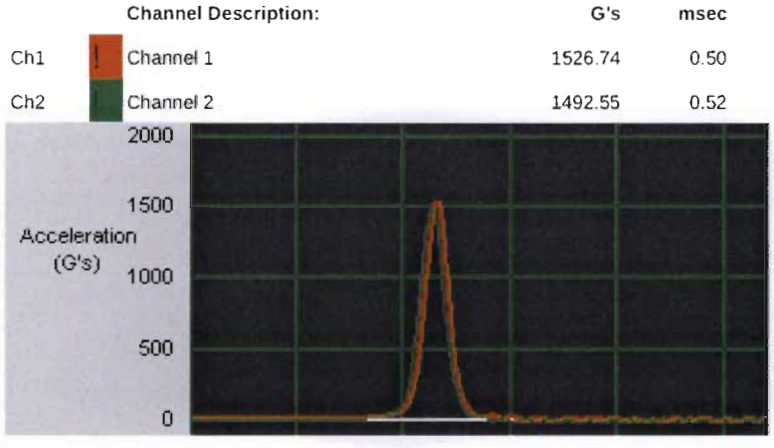

Figure 3.10: Input acceleration pulse of 1500G - 0.5ms, condition B

Because the paper impact surface gradually compacts over time the sheets of paper must be rotated or replaced after a number of impacts, otherwise the acceleration pulse will change. Table 3.1 shows the impact surface used for each acceleration condition and the empirically determined replacement rate for the paper sheets; the felt sheet was not replaced. The maximum error that was observed in either peak acceleration value or pulse width was less than 10%.

One deviation was made from the JEDEC standard in that the gap between the shoulder screw and board surface was controlled to within only 100 micrometers rather

Table 3.1: Acceleration pulse configuration

Acceleration Condition	Drop Height	Impact Surface	Replacement Rate (per test board)
900G-0.7ms	36.8 cm	1 sheet paper, 1 felt	1 sheet
1500G-0.5ms	57.2 cm	12 sheets paper	1 sheet
2900G-0.3ms	76.2 cm	6 sheets paper	1 sheet

than the standard 50 micrometers [1]. A misalignment of tooling for the drop test support screws prevented use of the specified gap limitation. As a consequence of this slightly looser restraint it is expected that the test vehicle would have a slightly larger deflection at the center during the impact test, and may have a reduced vibration damping coefficient. No detailed analysis of the effect of this change was performed.

The test vehicles were split into two groups as shown in Table 3.2, one per failure detection system, with each group having 8 edge bonded boards (4 for each edge-bond material) and 6 boards without edge bond. The groups were split so that each group would have 3 non-edge bonded boards at 900G, 3 non-edge bonded boards and 4 edge bonded boards at 1500G, and 4 edge bonded boards at 2900G. This setup provided a comparison between test systems with three acceleration conditions, and a comparison between edge-bonded and not edge-bonded components with two acceleration conditions. One additional board without edge bond was dropped at 2900G (no matching board was tested by post-drop); this board provided a rough comparison of edge-bond reliability improvement at 2900G but had no test replication. Prior testing suggested that the 2900G condition would cause failure very rapidly for the test board without edge-bond.

Table 3.2: Number of boards per drop test variable cell

Failure Detection	DAQ System		Post-drop System	
Edge-bonding	Yes	No	Yes	No
900G	0	3	0	3
1500G	4	3	4	3
2900G	4	1	4	0

3.3 Failure Detection Systems

This study compares drop impact failures with two failure detection systems: in-situ high-speed data acquisition (DAQ) with analog-to-digital conversion (ADC) yielding dynamic resistance measurement, and post-drop static resistance measurement. These systems will be referred to as the data acquisition system and the post-drop system for purposes of discussion.

For the DAQ system a cable was connected to the test vehicle by soldering the 16 wires (15 channels and common ground) directly into plated through-holes on the short side of the board. The cable was secured to the drop tester base plate to prevent loading against the solder connections during impact. The failure criterion used for the high-speed data acquisition system was taken directly from the JESD22-B111 standard [1]: a 100Ω resistance in the component daisy chain at any time during the drop impact or subsequent vibration is considered a failure, and the failure must be repeated in at least three out of five successive drops. Since a 100Ω static resistor and 5V DC supply voltage are used, the failure condition of 100 is the equivalent of measuring 2.5V on the component daisy chain. The nominal initial resistance of the daisy-chains was between 1Ω and 3Ω depending on component location on the board.

The electrical continuity of the cable-to-board through-hole solder joints was verified before testing, and at regular intervals during and after drop testing to eliminate false positive failures due to broken cable connections. The false positive condition

occurred several times during the test process and was much more likely to occur for the higher acceleration conditions, especially with the 2900G condition. To prevent broken cable connections during the prolonged testing on the 2900G condition a foam pad was placed underneath the edge of the cable connection, and was taped to hold it in place. This allowed the wires to bend gently over the foam pad and not to flatten to a 90 degree bend at the solder joint. However, this also created a slight mass increase on the board on the cable end, due to the foam pad and tape, as shown in Figure 3.11. The additional mass on the board from this foam and tape is minimal, but may influence board deflection during and after impact. Future drop testing with a data acquisition system should establish a method of preventing the sharp 90 degree wire bend which was responsible for some solder connection failures, and slowed down testing considerably. Unfortunately simple connectors are inadequate for this job as previously mentioned. Any system of cable attachment should be mirrored to both ends of the board to minimize the effects of imbalanced mass which may change the board deflection shape as discussed in Section 4.4.

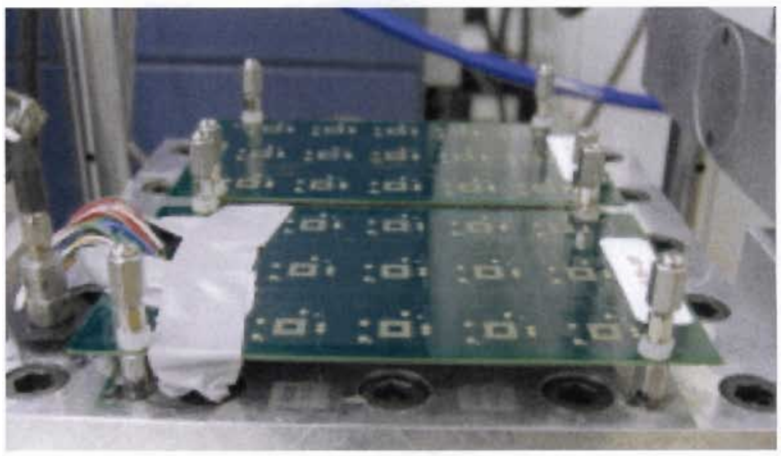

Figure 3.11: Test vehicle and DAQ system cable with taped foam support

The post-drop resistance measurement system uses a LabView program and Keithley digital multimeter to read the daisy-chain resistance, once for each component after each drop, through a cable connected to the test vehicle after the vibration

ceases. For this system no cable is connected to the board during the drop event, so it is not an in-situ failure detection system. The failure criterion for the post-drop resistance measurement system is a static 10Ω rise (or more) from initial resistance.

The two failure detection systems use failure criteria that are necessarily different. The post-drop system is detecting a class of permanent failures only. That failure may be a solder crack partially seated together when the board is at rest; the change in the resistance of the daisy chain due to this crack is small. The 10Ω static rise threshold was chosen to detect that small change in resistance with the board at rest. The data acquisition system is designed to detect intermittent failure. That failure may have insignificant resistance change when the board is at rest but a larger change during board deflection, and may be an early indication that a solder crack will soon form. The data acquisition system uses a temporary 100Ω resistance threshold to indicate failure, but also may be used to identify early transitional failure. Data samples taken by the data acquisition system after the board vibration ceases could also be used to detect failure with the post-drop criteria.

The high-speed data acquisition system is capable of detecting intermittent failures as shown in Figure 3.12 during the board deflection and vibrations, whereas due to the single static measurement taken per drop the post-drop system can only detect permanent failures. In the example shown the static resistance rise from the initial condition is negligible and would not be detected as failure by the post-drop system since the static resistance rise is less than 0.1 ohm from the initial condition.

Additional examples of the DAQ system output can be found in Appendix B. The three classes of failure condition used in this study, transitional failure, intermittent failure, and complete failure, are shown. For purposes of determining drops until failure the transitional failure condition was not included, so only when intermittent or permanent failure occurred was a component considered failed.

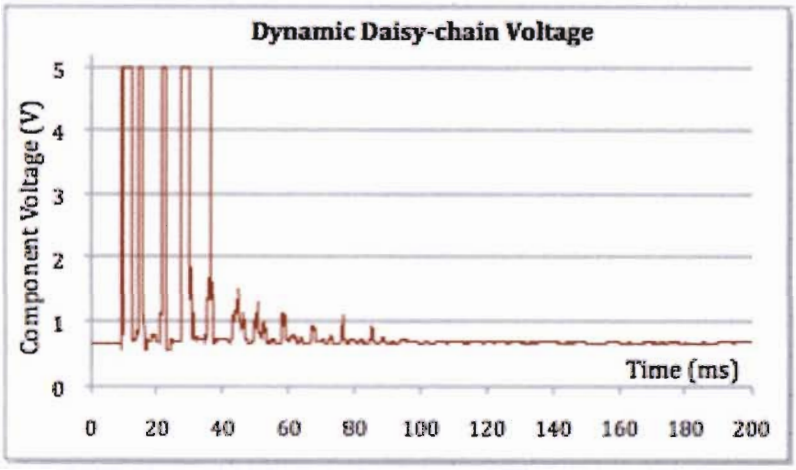

Figure 3.12: Intermittent failure detected by DAQ system, 10,000 data points shown in a 0.2 second window

Chapter 4

Results and Discussion

4.1 Observed Drops Until Failure

Because the JEDEC standard drop test induces a complex strain pattern across the test board, causing varied stresses in the solder joints, JEDEC recommendations divide the components on the board into six groups (denoted A-F) that are expected to have similar failure rates due to the symmetry of their locations [1]. The issue of component location has been shown in a number of studies to be critical; the stress and strain in solder joints, and their failure rate, is partially dependent on the component location on the board [22, 23]. Che, et al. found that the maximum acceleration location occurs at the board center and is much higher than the input acceleration, however the maximum board strain occurs under components along the board edges and near the supports [22]. Therefore it is necessary to discuss failures in context of component location. The component locations are numbered as shown in Figure 4.1. Note that for this study the DAQ system cable wires are always soldered in plated through-holes (PTH) at the board edge near component C6; the location of this cable and its influence on reliability results is discussed in the following sections.

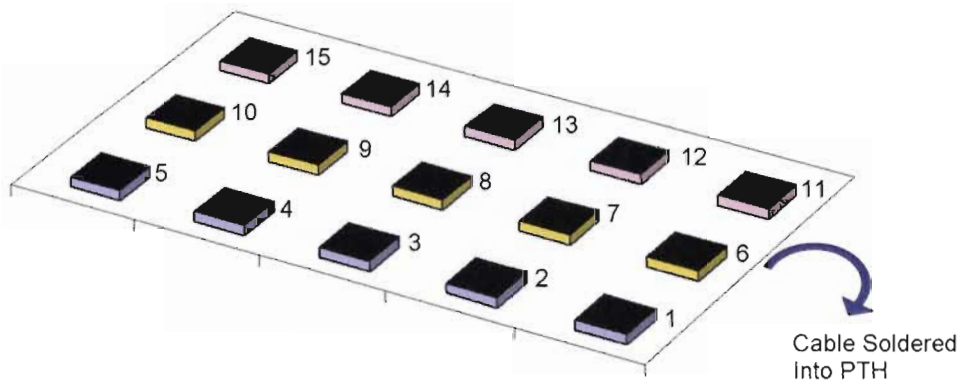

Figure 4.1: Board component location numbers for 15 components

4.1.1 CSP reliability without edge bonding

The drops to failure for each component location and test board without edge bonding are shown in Table 4.1 for the DAQ system and in Table 4.2 for post-drop system. In both tables, each column represents one board except the first column. The first row is the input acceleration condition used for that board and the second row is the total number of drops the board was subjected to.

It is clear that the drops-to-failure is different between the DAQ system and the post-drop system. One of the most obvious differences is that component C5 failed for every post-drop system board at 900G and 1500G but not at all for the DAQ system for those conditions. C14 and C9 also have similar differences between the detection systems, both failing more for the post-drop system than data acquisition. The reason for this difference in failure between systems is not clear, however it may be due to the attachment of the data acquisition cable to the board, which is believed to have an effect on board dynamics.

The component location plays a significant role in the drop test reliability. The components along the board center (C3, C8, C13) tend to fail earliest and most frequently for both failure detection systems, although C3 did not fail as often for the

DAQ system. Components C4 and C12 also failed consistently for both systems, but the symmetrical board locations of C2 and C14 did not fail as often or as early; C2 did not fail consistently for either system, but C14 did fail for the post-drop system. The failure locations have other symmetry issues as well; with C9 failing on the post-drop system but the symmetrical match C7 failing infrequently.

Table 4.1: Drops to failure by data acquisition without edge-bonding

Accel (g)	900	900	900	1500	1500	1500	2900
Drops	75	75	100	70	40	60	50
Component							
C1				37	29		7
C2							25
C3	62				14	33	4
C4	26	26	34	26	6	23	4
C5							5
C6					21	35	3
C7					19		42
C8	28	44		50	3	13	7
C9					30		21
C10							
C11					5		11
C12	16	6	43	13	2	6	4
C13	15	11	40	9	1	5	2
C14					21	32	38
C15							50

It is interesting to note that the drops-to-failure vary significantly between different boards for the same component location. It is clear that higher G-level results in lower drops-to-failure. Every component except C10 in a board without edge-bonding failed after 50 drops when subjected to 2900G. Most of the components fell off the board after less than 20 drops.

Table 4.2: Drops to failure by post-drop without edge-bonding

Accel (g)	900	900	900	1500	1500	1500
Drops	75	70	100	70	40	60
Component						
C1			82	55		38
C2						22
C3	7	31	15	8	3	11
C4	10	43	17	7	5	36
C5	65	2	14	1	5	14
C6	54					45
C7			61			9
C8	13	13	16	7	5	2
C9	53	16	11	28	8	14
C10						
C11	29		55			12
C12	6	9	18	5	3	3
C13	5	28	16	5	3	3
C14	1		37	5	34	4
C15	44		75	26		

4.1.2 CSP reliability with edge bonding

The drops to failure data for edge-bonded boards are reported in Tables 4.3 and 4.4. The total number of drops for each board is listed in row 2, and the edge bonding material (either thermal-cured epoxy or UV-cured Acrylated Urethane) is listed in row 3.

It is clear that edge-bonding improves the drop test reliability significantly by comparing the highlighted columns in Table 4.3 (2900G) to the last column of Table 4.1 (also 2900G). Eight components failed on a board without edge-bonding after 7 drops when subjected to 2900G as shown in Table 4.1, while the first eight failures occurred for boards with edge-bonding after 36, 44, 100, and 133 drops when subjected to 2900G as shown in Table 4.3. For an input acceleration of 2900G, the edge-bonded boards show a 5-8 times reliability improvement. The component location plays a significant role in the drop test reliability. Similarly to the boards without edge

bonding, components C4 and C12 fail earlier than components C2 and C14, in the symmetrically mirrored board locations. Again as with the boards without edge bonding, components C7 and C9 show significant symmetry mismatch in both board failure detection systems. This issue is explored further in the failure analysis section where the reason for this difference is explained.

The drop counts to failure are higher with edge bond applied for the majority of boards and component locations, and for both failure detection systems. The data acquisition system observed some intermittent failures that occurred for up to 150 consecutive drops in edge-bonded components without ever advancing to a permanent failure stage. In some of these cases the post-drop system would not have recorded failure when drop testing was stopped.

Table 4.3: Drops to failure by data acquisition with edge-bonding

Accel (g)	1500	1500	1500	1500	2900	2900	2900	2900
Drops	325	350	279	355	190	170	175	173
Component								
C1						151	66	61
C2		342	276		133	127		119
C3	80	292	33	101	70	72	12	103
C4	236	255	257		63	16		100
C5						36	73	91
C6		55				44	37	60
C7						35	69	158
C8	201			85	113	20	84	83
C9				292		25	29	124
C10			277			12	59	
C11		193	178	103		65	38	
C12	66	76	52	162	53	24	23	16
C13	61	129	73	77	42	13	18	14
C14		232				42	44	120
C15	107		268		44	22	25	90

The data acquisition system does not always show failures in fewer drops (earlier detection) as was expected since it can detect the intermittent failure, but it recorded more total failures of the 2900G set than the post-drop system did. The capability

Table 4.4: Drops to failure by post-drop with edge-bonding

Accel (g)	1500	1500	1500	1500	2900	2900	2900	2900
Drops	237	350	279	300	170	170	175	173
Component								
C1		304	62			12	23	
C2			101				34	98
C3	2		180	81	74	72		23
C4	2	292	99	242		25	13	
C5	60		62	262		40		151
C6	112	282	180			151		
C7		6						
C8	88			108		68	30	21
C9		132		283	116	106	53	
C10		112						
C11	3	292				112		
C12	1	36	188	162	137	57	154	128
C13	159	99	188	133	6	144	36	43
C14	60			243			151	
C15				297				

of detecting failure earlier may be partially offset by the requirement of adding wired connections to the board during the drop impact; the wire may influence board deflection and vibration characteristics, and subtly effect drop reliability results. This issue is further explored in Section 4.4.

4.2 Failure Analysis

Failure analysis was performed on a subset of the failed test boards after drop tests. The outer row of solder joints of two components on two boards were cross-sectioned. Scanning electron microscopy (SEM) images indicate the intermetallic layer thickness was 1-1.3 micrometers on the board side and 1.3-2 micrometers on the component side. To investigate the extent of cohesive failure resulting from the drop tests, the dye penetrant test was performed on eight boards, four with and four without edge-bond. Optical microscopy was used to identify dyed areas and determine failure location,

root cause, and how widespread the under-pad resin cracking problem was for each component location.

4.2.1 Failure Modes

The most common failure observed was trace/pad breakage at the neck from the trace to pad as shown in Figure 4.3. The dyed area in the right of Figure 4.3 shows the resin cracked under the copper pad on the board side. The trace break was mainly due to the cohesive failure of resin between the copper pads and the fiberglass dielectric layer. Figures 4.5-4.7 show cross-sectioned solder joints where resin cracking is visible underneath the pads in the dielectric layer. The pad cracking was commonly seen for both boards with and without edge bonding. Similar failure modes with broken traces have been reported by Chong, et al. [24], and resin cracking has been observed by Mattila, et al. [16], Chong, et al. [24] and Wong, et al. [25].

All the components that electrically failed and were examined by the dye penetrant method were categorized as solder failure, input/output (I/O) trace failure, or daisy-chain trace failure. The I/O trace and daisy-chain trace failures only occurred when pad cracking was present.

Of those components with electrical failure: 58% showed I/O trace failure with cracking under pads, 12% showed solder joint fracture on the board side, 19% showed both solder joint fracture and I/O trace failure with cracking under pads, and the remaining 11% showed daisy chain trace failure with cracking under pads. Figure 4.2 shows these values graphically. Where both solder fracture and I/O trace failure was observed it was impossible to determine which occurred first.

The group of 58% of electrical failures were due to pad cracking under one or two I/O trace connections that caused the trace to be broken away from the copper pad. Another 11% had under pad cracking that led to daisy-chain trace failure, a trace

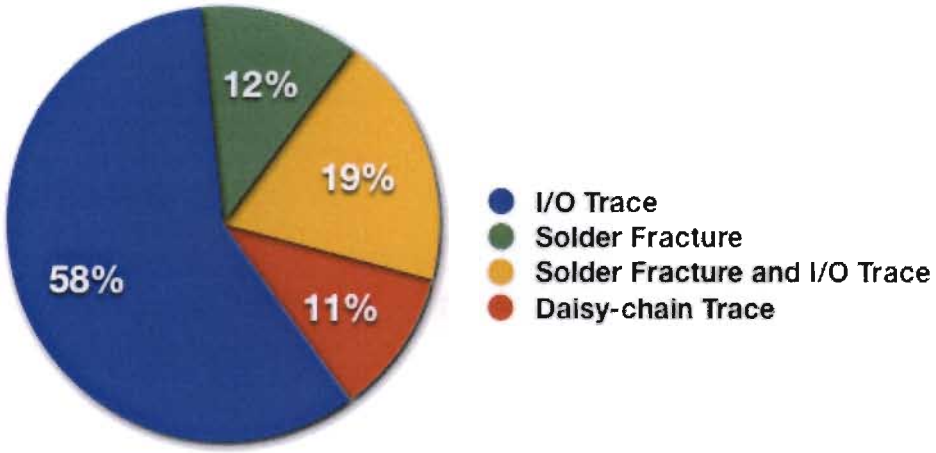

Figure 4.2: Electrical failures classified by cause

broken between two copper pads, within the ball-grid array rather than at the I/O traces. Both trace failure types were caused by under pad cracking which led to the copper pads separating away from the PWB. The traces adhered to the board surface were broken when the copper pads separated from the board.

The large ratio of electrical failures resulting from I/O traces cracked away from pads compared to solder joint fracture may be partially related to the test vehicle design and trace routing. Figure 4.8 shows a single CSP pad location with all four corners numbered. Corner 2 has two traces running outward from the component. These two traces are the daisy-chain input and output connections. Traces connected to the other three corners lead to test pads and are not part of the daisy chain; connectivity to these test pads was not sampled by the DAQ system during testing, instead only continuity of the daisy-chain through the two I/O traces was sampled. The orientation of every package on the test vehicle is the same, with corners 1 and 2 parallel to the short board axis, and corners 2 and 3 parallel to the long board axis; the orientation of the trace layout is also the same for each component. Due to resin crack under the copper pad, the transition of the trace to the I/O pad is the weakest point, which causes copper trace/pad cracking. A copper pad crater and part of one

of the corner 2 I/O traces is shown in Figure 4.9, which is the board side match of the failure shown in Figure 4.3.

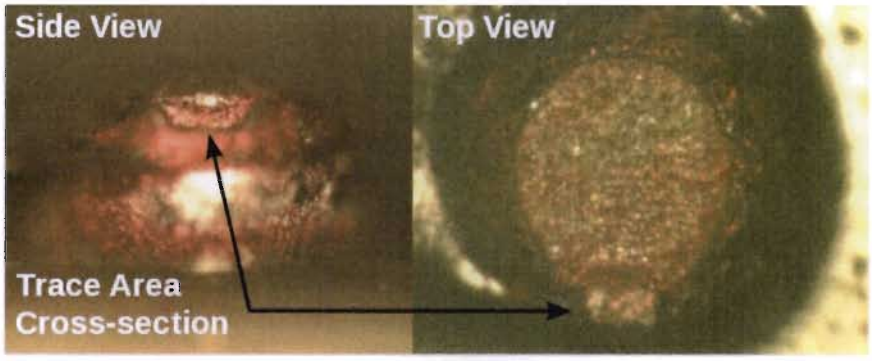

Figure 4.3: Trace cracked away from solder joint (left) and the same solder joint with pad dyed (view of component surface)

The frequency of the pad cracking caused failures indicates that the drop impact reliability of the assembly might be overestimated in some circumstances. The current test vehicle uses traces to connect from a copper pad directly to the neighboring copper pad. If the copper pads lift off the board while maintaining electrical continuity, as when the daisy-chain trace between copper pads lifts up with CSP, then electrical failure would not be detected when it should have occurred. A test vehicle utilizing a more typical PWB layout for CSPs in electronic devices may be more appropriate for evaluating board level drop impact reliability. A typical layout would create the daisy-chain connections by routing traces from copper pads to vias, or using vias in pads, rather than directly routing traces from pad to pad. That layout would not allow daisy-chain traces to lift off the board while still making electrical connections. Figure 4.4 shows an example of both pad-to-pad and pad-to-via trace routing for a typical BGA. In both examples the black segments are connections made within the CSP itself, the copper segments are external traces on the PWB, and the green segments are connections made in the PWB internal layers.

The secondary failure mode was solder joint fracture. Figure 4.10 shows a fracture

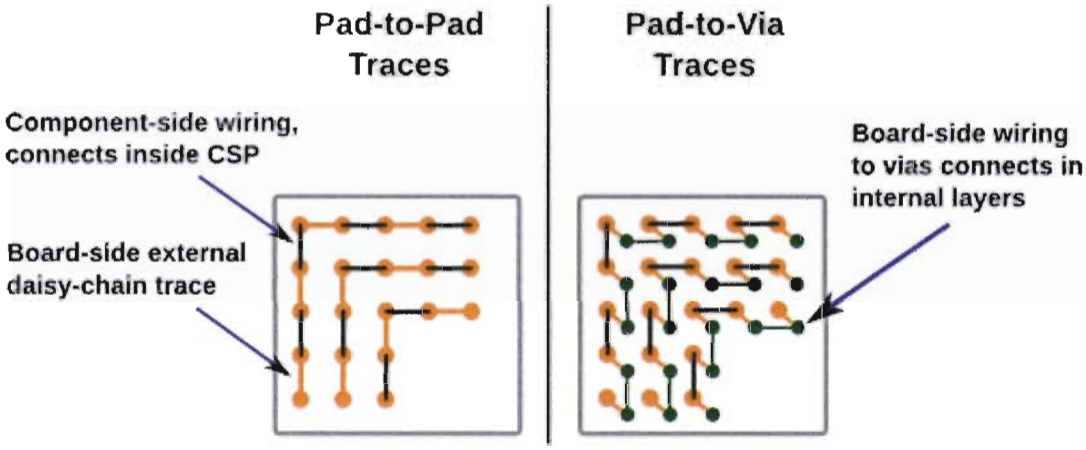

Figure 4.4: Example of copper pad-to-pad and pad-to-via routing for a common BGA

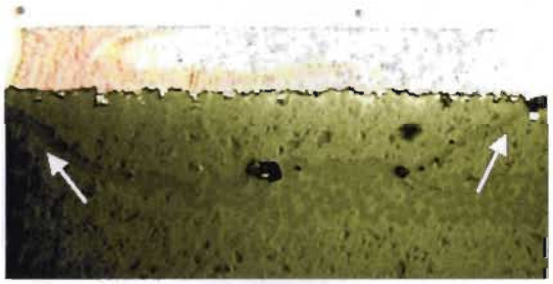

Figure 4.5: Cracked resin under the board side pad (dark line), edge bonded

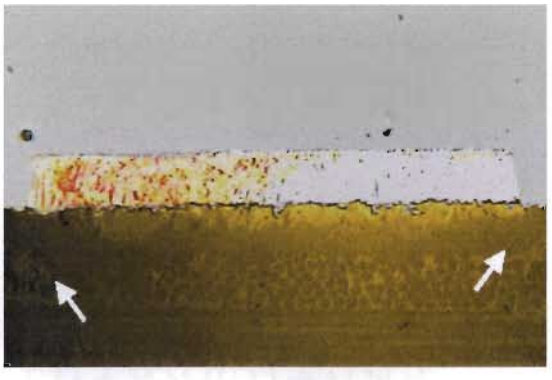

Figure 4.6: Crack in board resin underneath pad (thick dark area), no edge bond

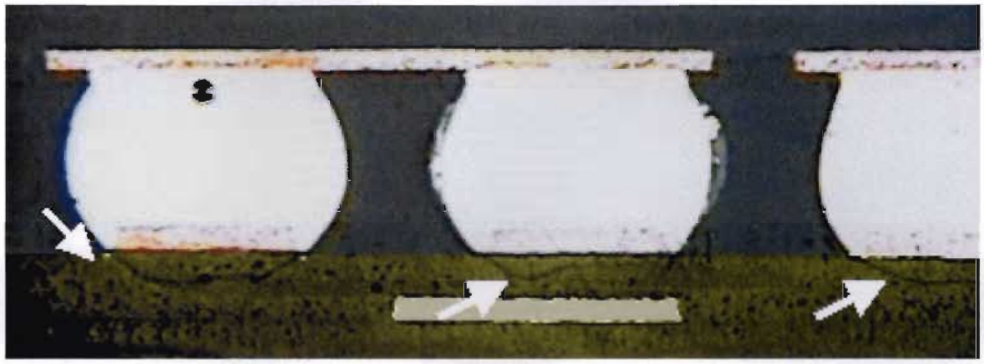

Figure 4.7: Cracked resin layer under pads for several solder joints, edge bonded

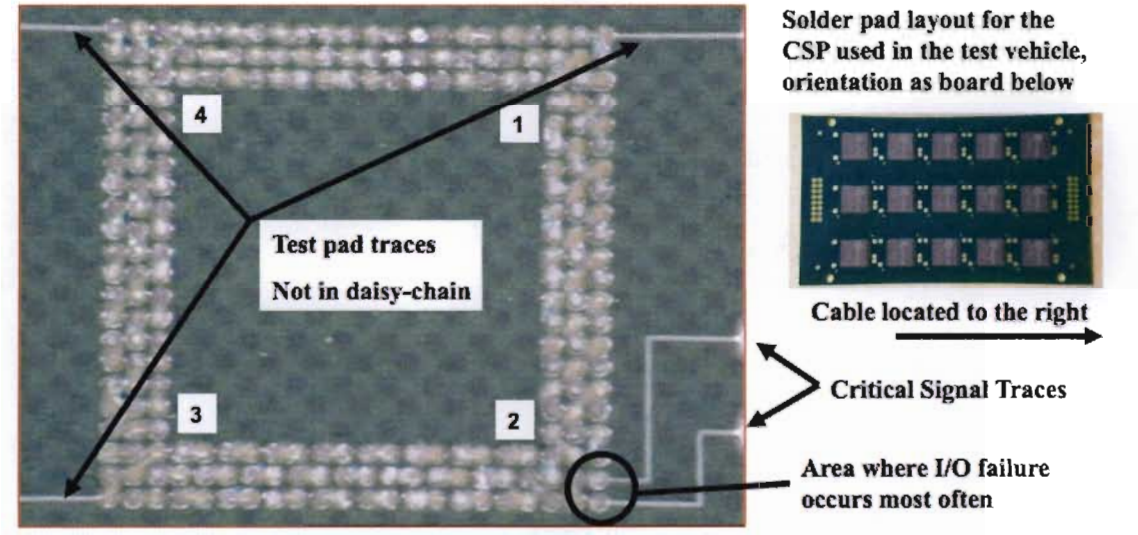

Figure 4.8: CSP I/O traces and component orientation

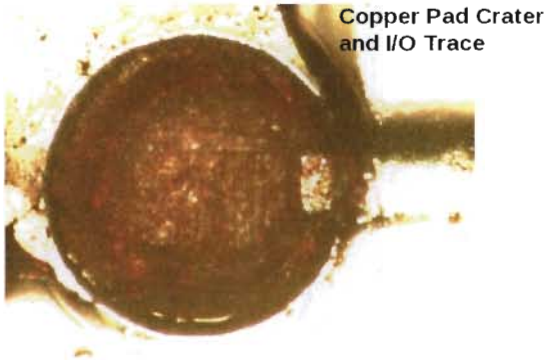

Figure 4.9: Copper pad crater showing dyed board fibers

near the board side Cu_6Sn_5 intermetallic layer. Solder fracture failures were observed at the board side primarily, and only one solder failure was found at the component side. Figure 4.11 is an illustration of the failure locations observed in the study, either pad cratering, board side solder fracture, or a combination of the two. Both complete and partial solder fractures were found by dye penetrant analysis as shown in Figure 4.12. It is interesting to note that both a solder joint fracture and a broken trace can lead to electrical failure as shown in Figure 4.13. The pad on the left side of the image in Figure 4.12 has resin cracking which led to trace breakage as the pad lifted away from the board with the component during board deflection.

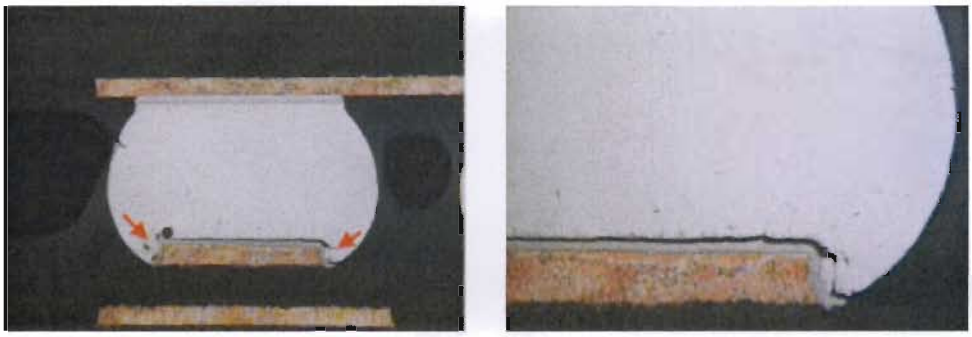

Figure 4.10: Solder joint fracture near the board-side IMC layer

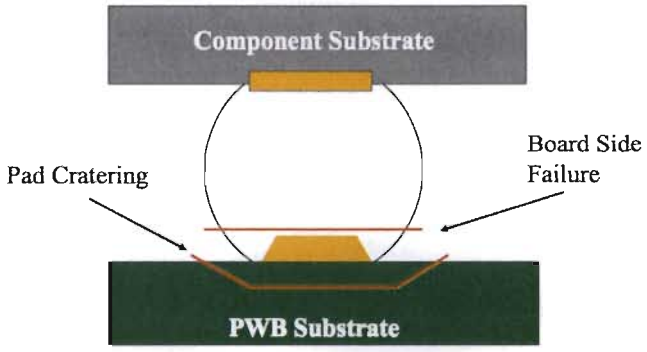

Figure 4.11: Illustration of failure locations observed in solder interconnects

The extent of resin cracking under the copper pad was examined for all components of the eight boards that were dye penetrant tested. Out of these 120 components

Figure 4.12: Complete fracture (left) and partial fracture (right)

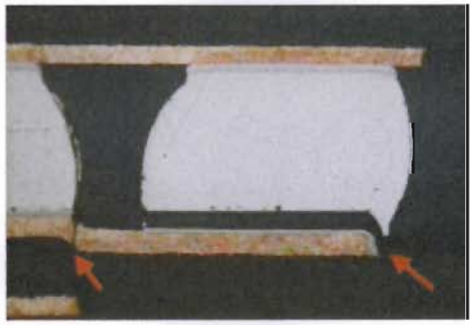

Figure 4.13: Board side solder fracture and broken trace

(15 components per board times 8 boards), the relationship between electrical failure and resin cracking is summarized in Table 4.5. Table 4.5 shows that 72% of components that electrically failed had some resin cracking under the copper pads, while 19% of components that had not electrically failed had resin cracking. The remaining 9% of components did not have resin cracking. This indicates that the solder joints are not the weakest link area of the assembly. Therefore, this research supports the recommendation that board laminate materials be improved.

Table 4.5: Relationship between electrical failure and resin cracking

		Electrical failure	
		Yes	No
Resin cracking under pads	Yes	72%	19%
	No	6%	3%

4.2.2 Differences between edge bond material failures

There are notable differences in the mechanical failure mode between the two edge-bond materials. The epoxy material tends to fracture through the edge-bond material as shown in Figure 4.14. More than 20 components that were edge bonded with the epoxy material, or more than 10% of all the components in the group, dropped off the board during testing. This fracturing was observed to occur before electrical failure happened. The acrylic edge-bond material did not fracture, but delaminated from the package sides. The acrylic did not delaminate from the board surface. Figure 4.15 shows that four undamaged edge bonds remained on the board after the component fell off. The properties of these two edge-bond materials are believed to contribute to the difference in the mechanical failure mode.

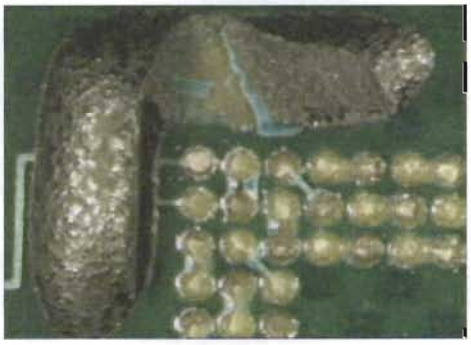

Figure 4.14: Fractured thermally cured epoxy edge bond

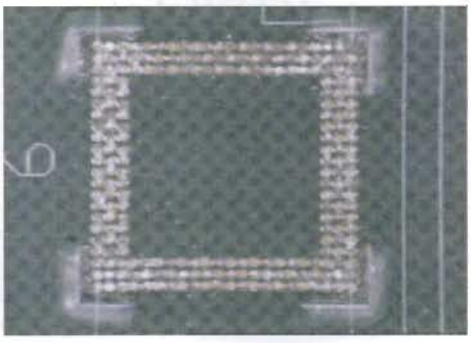

Figure 4.15: Four UV cured acrylic edge bonds after CSP failure

4.3 Pad Cratering and Solder Fracture Maps

The failure type and locations of failures were mapped on four test vehicles that were examined by the dye penetrant method. These maps provide a view of how common the cracking under pad failure type is on these test vehicles and for which component locations it is most prevalent. For each of the fifteen components on these four test vehicles there are 228 squares representing each solder interconnect in the perimeter BGA. All four test vehicles mapped here were drop tested using the data acquisition system, so the cable was attached near component 6 during each drop impact.

Figures 4.16-4.19 each show the component maps for one test vehicle. In each figure the color filled in the squares represent the failure mode: pad craters are black, solder fractures on the board side are red, solder fractures on the component side are yellow, and white squares are not-failed solder interconnections. There are very few red squares, and only one yellow square (in component 11 in Figure 4.17); the vast majority of solder connections either had no failure or showed pad cratering. Recall that electrical failure does not occur for all solder interconnections that showed pad cratering because the daisy-chain traces may still be connected despite being cracked away from the PWB. When either board or component side solder fracture is observed there was no pad cratering present under those copper pads.

The failure maps are valid only for that test vehicle, at the specified acceleration condition and number of drops; since each is different general conclusions may not apply broadly to other test vehicles. It is informative however that pad cratering is typically seen at the component corners after very few drops (Figures 4.18 and 4.19). The failure maps for test vehicles with edge-bonded components show that after many drops several components have no pad cratering, while for the test vehicles without edge-bonded components nearly every component has pad cratering after few drops.

This reinforces the conclusion that edge-bonding is effective at reducing pad cratering.

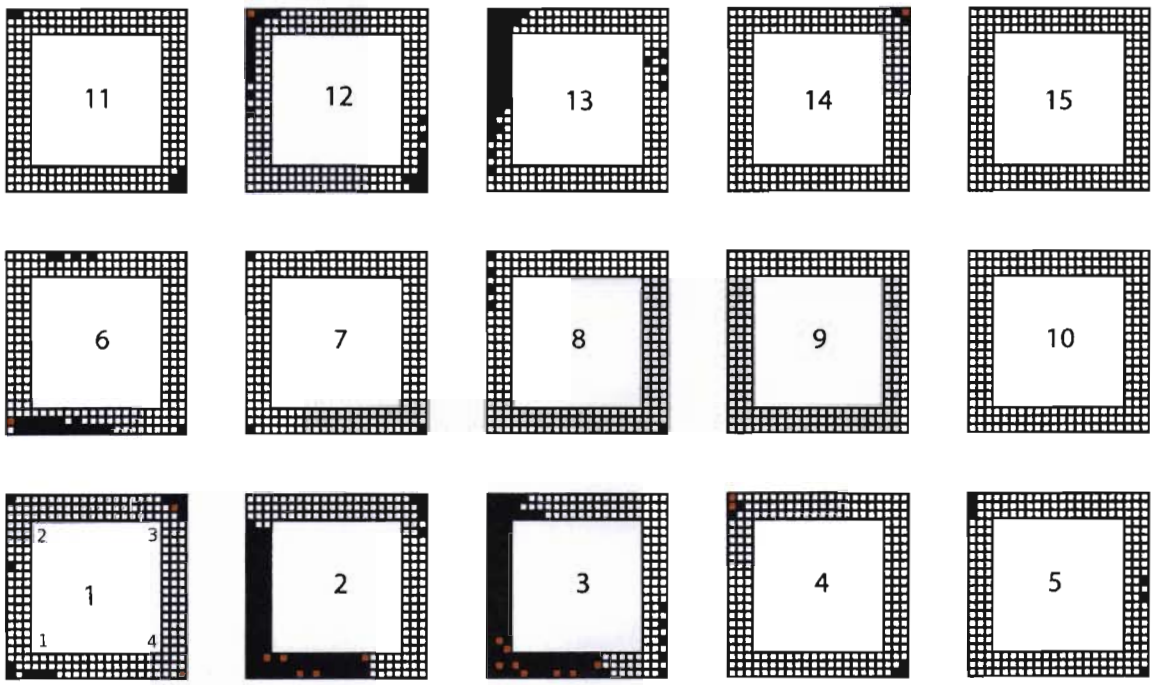

Figure 4.16: Failure map for epoxy edge-bonded CSPs after 325 drops at 1500G

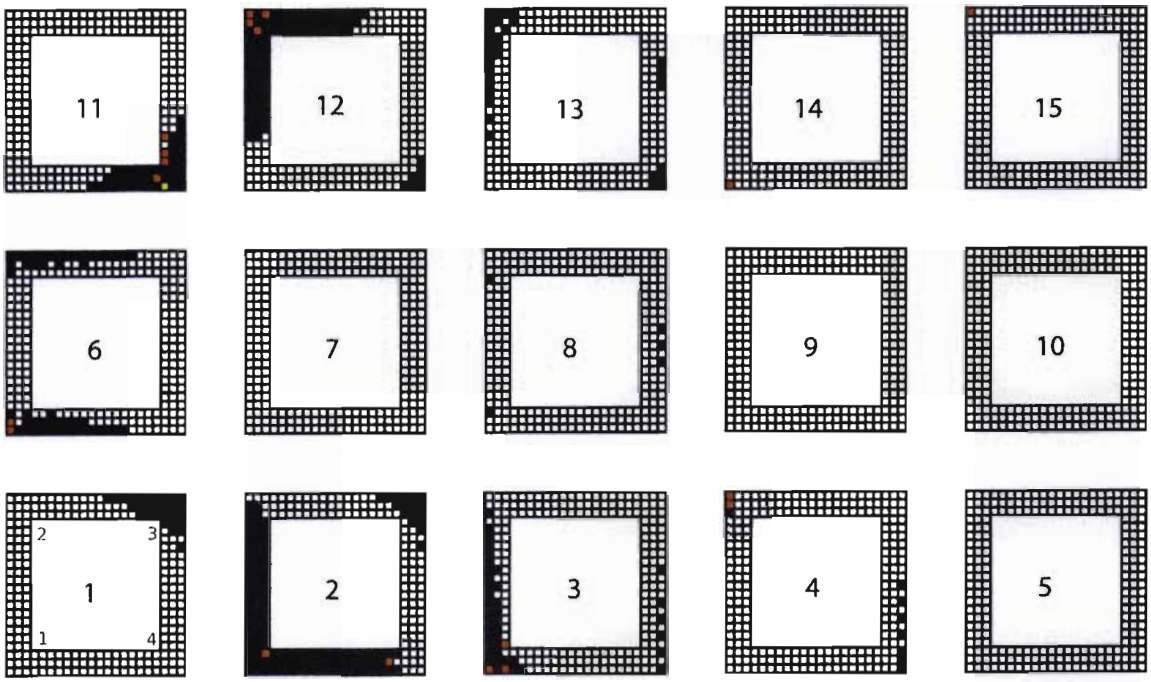

Figure 4.17: Failure map for acrylic edge-bonded CSPs after 279 drops at 1500G

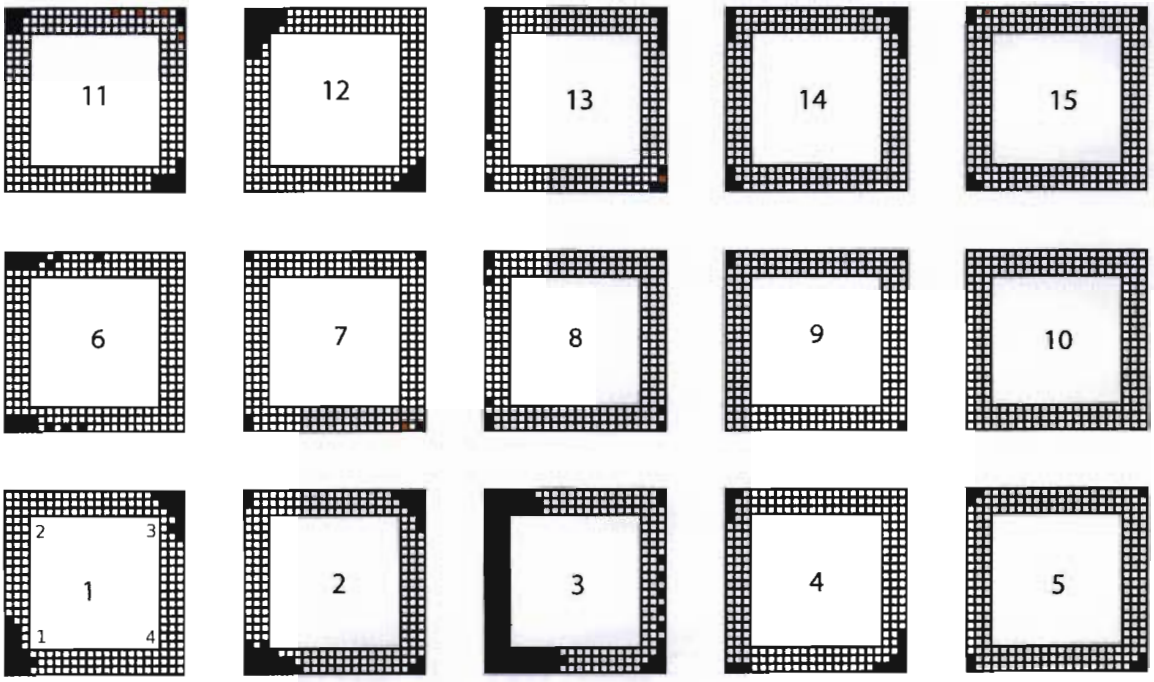

Figure 4.18: Failure map for not edge-bonded CSPs after 10 drops at 1500G

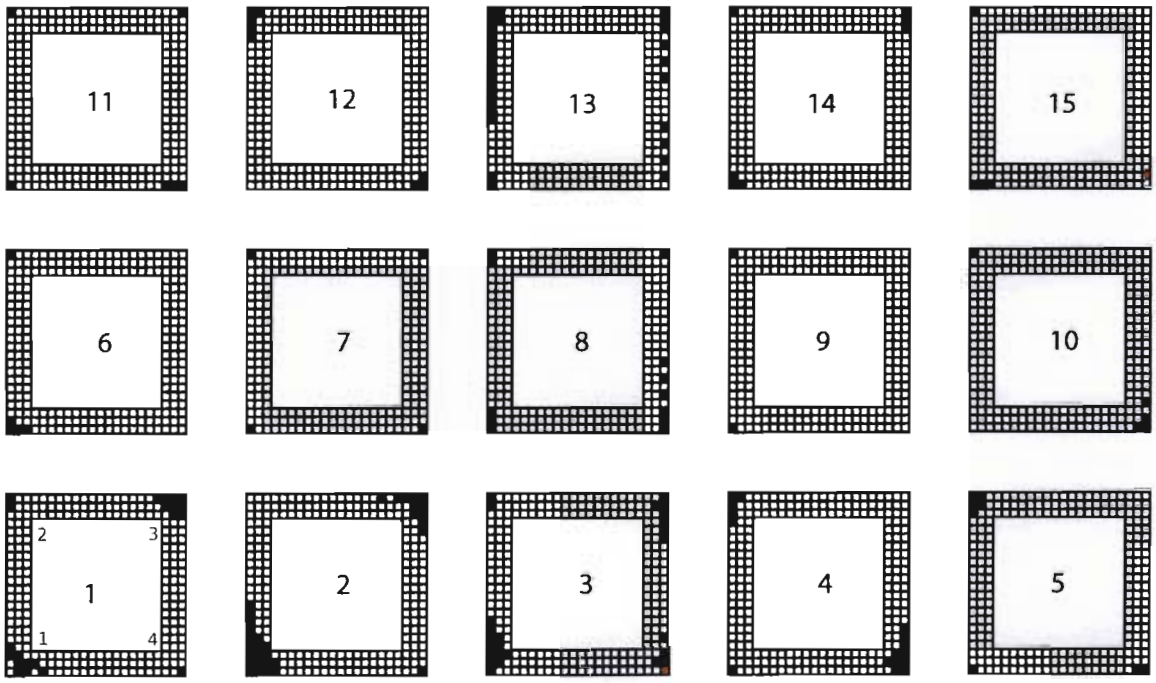

Figure 4.19: Failure map for not edge-bonded CSPs after 14 drops at 900G

4.4 Acceleration Peak Values

The acceleration profile on the test vehicle is not the same at all component locations. To better understand how this effects drop impact reliability, observations of the acceleration at each component location were taken. Four test vehicle assembly conditions were tested: with no components (blank), without edge-bonding, with epoxy edge-bonding, and with acrylic edge-bonding. For each of these conditions two observations were made at each component location, and the average of the two observations was recorded. Using two accelerometers simultaneously the acceleration at the component location was observed and compared to the acceleration at the drop table, which is the input acceleration pulse. For each observation the input acceleration applied to the drop table was 1500G with 0.5 millesecond pulse width, JEDEC Condition B [1].

The acceleration observed on the test vehicle is not a single half-sine impulse

because the test vehicle vibrates after the impact and has acceleration during this cyclic motion. The peak acceleration of the drop table is typically not observed on the test vehicle during the timeframe of the original half-sine input acceleration pulse, but instead the peak acceleration value at the test vehicle occurs during the vibration from 0.5 to 3.0 milliseconds later. As shown in Figure 4.20 the shorter, red color, half-sine pulse is the input acceleration of 1500G and 0.5 millesecond duration. The green line shows the acceleration values at the center of the test vehicle, at component location C8; the test vehicle being recorded in this figure has no edge-bonding and no DAQ cable attached (this test matches Figure 4.25). The peak acceleration value is observed to occur almost one millesecond later, and has a peak value of approximately 2400G and pulse duration of 0.4 millesecond.

The average of two acceleration values for each test vehicle condition are reported in Table 4.6, and each value is plotted in Figures 4.21-4.25. The first column shows the component number which also indicates location on the test vehicle. Each other column represents one test vehicle condition, which is a combination assembly status and edge-bonding. The columns marked with 'C' have the DAQ system cable attached to the test vehicle while columns marked with 'NoC' do not have the cable attached. Columns marked with 'NoEB' have no edge-bonding, while columns with either 'Epoxy' or 'Acrylic' indicate either the thermal cured epoxy or UV-cured acrylic edge-bonding material was used.

The results of these recorded peak acceleration values shows that when the DAQ system cable is attached to the test vehicle it influences the acceleration peak values observed on the board during the test. The DAQ system cable is attached to the test vehicle near component C6 (refer to Figure 4.1). With the cable attached there is a loss of symmetry between the C6 and C10 locations, as well as between the C7 and C9 locations. Likewise there is a typical loss of symmetry between locations C1 and C11, and between locations C5 and C15. The maximum peak value is also effected

Table 4.6: Average acceleration peak value for test vehicle variations with 1500G 0.5-ms input acceleration

	Peak Acceleration Value			
Component	Blank C	Blank NoC	NoEB C	NoEB NoC
C1	1642	1668	1688	1540
C2	1049	1501	1405	1640
C3	1395	2556	1309	1508
C4	1244	1482	1411	1493
C5	1236	1270	1291	1201
C6	1906	2003	1406	1821
C7	2392	1895	2084	2149
C8	2458	3492	2202	2443
C9	1947	2218	1588	2034
C10	1849	1898	1905	1890
C11	1794	2083	1503	1498
C12	1121	1581	1090	1383
C13	1547	2521	1142	1563
C14	1168	1405	1401	1559
C15	1114	1381	1182	1271

	Peak Acceleration Value			
Component	Epoxy C	Epoxy NoC	Acrylic C	Acrylic NoC
C1	1576	1264	1750	1253
C2	1555	1564	1418	1504
C3	1168	1431	1184	1258
C4	1284	1337	1252	1443
C5	1187	1020	999	1026
C6	1322	1513	1418	1571
C7	1969	1914	1846	1877
C8	2150	2288	2204	2403
C9	1585	1634	1575	1587
C10	1780	1471	1865	1399
C11	1209	1239	1377	1235
C12	1323	1343	1299	1374
C13	1397	1601	1232	1499
C14	1200	1475	1242	1454
C15	1088	1260	1175	1134

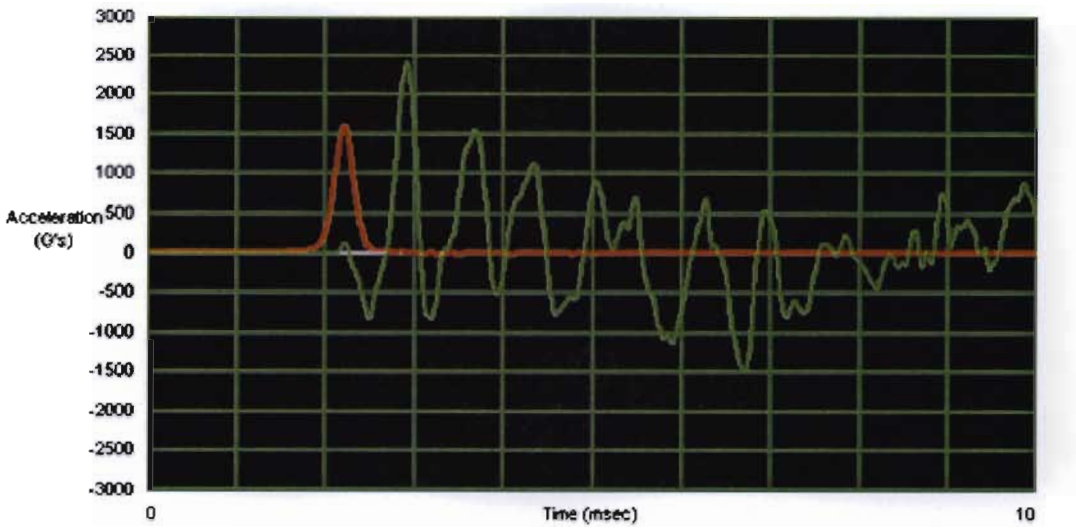

Figure 4.20: Real acceleration versus time for 1500G-0.5ms input pulse

by the presence of the DAQ cable, which is most noticeable for the test vehicles with no components (blank) and with no edge-bonding. The effects of the DAQ cable on the test vehicle's dynamic response is less severe with the edge-bonded components, most likely due to stiffening of the PWB by the application of edge-bond material, but no tests were conducted to verify this assumption.

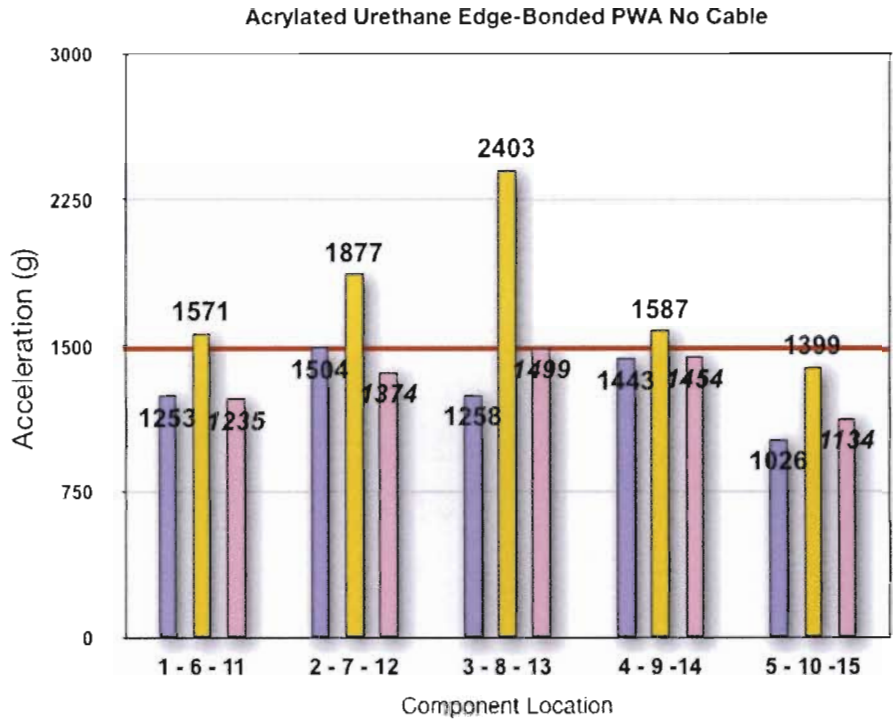

Figure 4.21: Accelerations on acrylic edge-bonded CSPs without DAQ cable

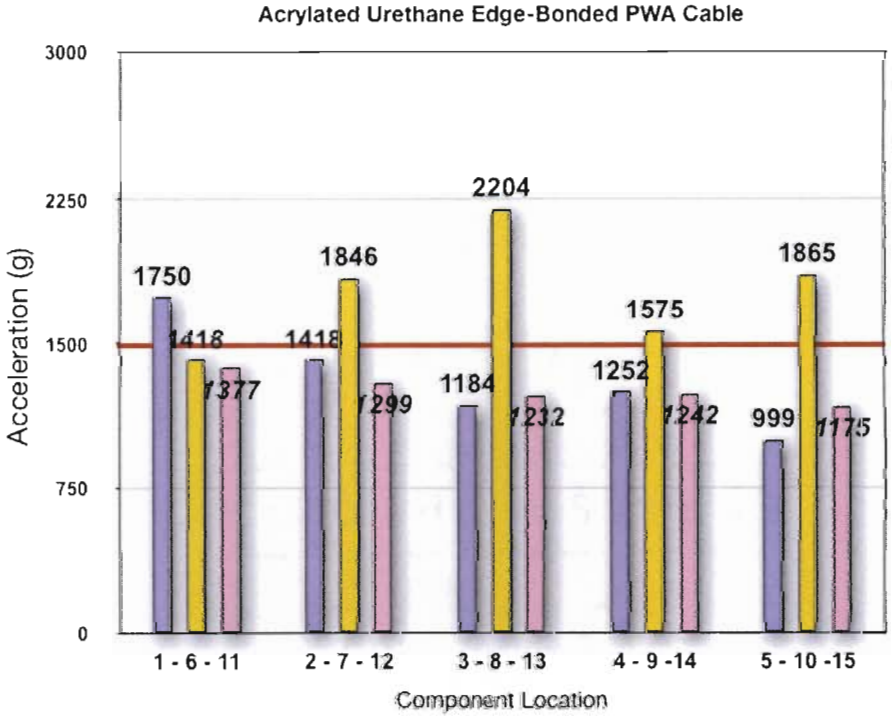

Figure 4.22: Accelerations on acrylic edge-bonded CSPs with DAQ cable

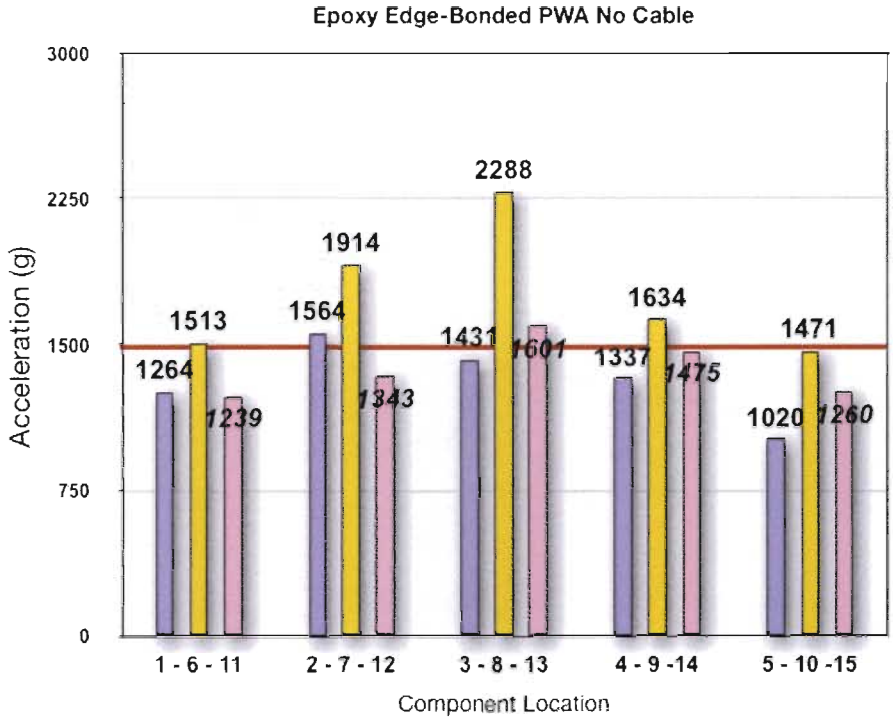

Figure 4.23: Accelerations on epoxy edge-bonded CSPs without DAQ cable

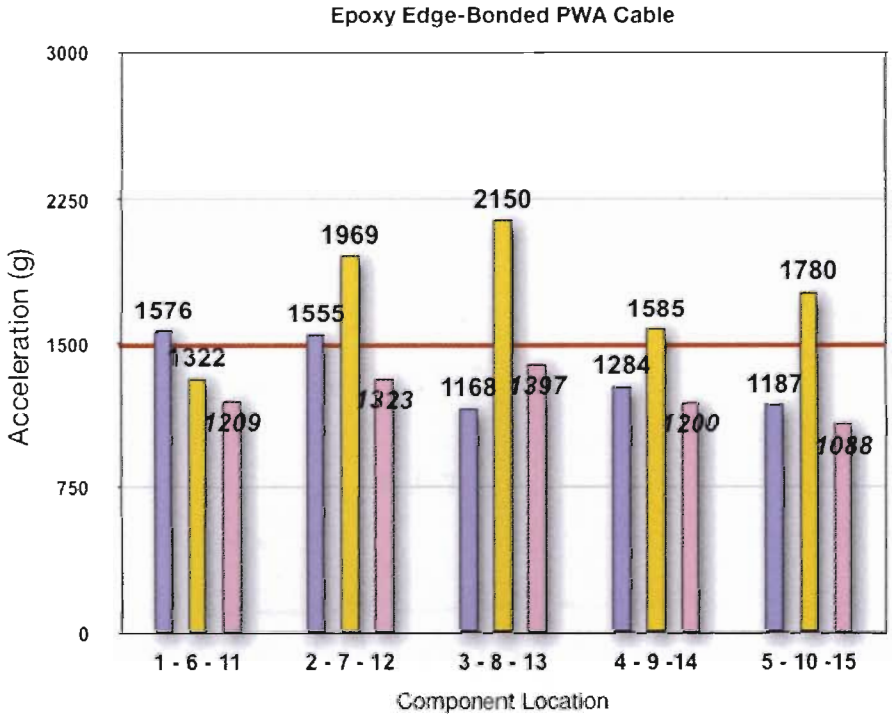

Figure 4.24: Accelerations on epoxy edge-bonded CSPs with DAQ cable

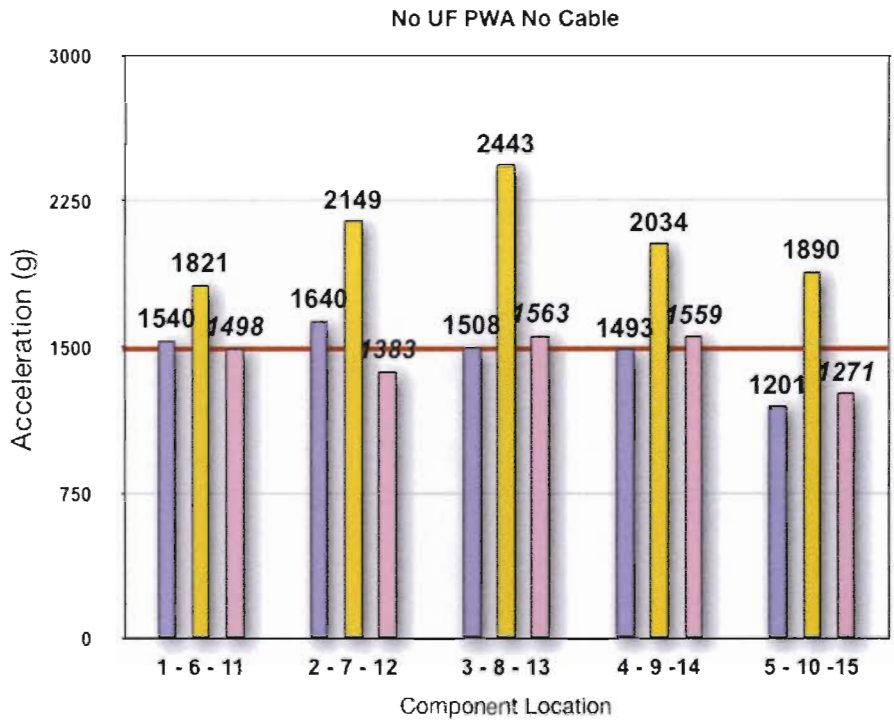

Figure 4.25: Accelerations on CSPs without edge-bond without DAQ cable

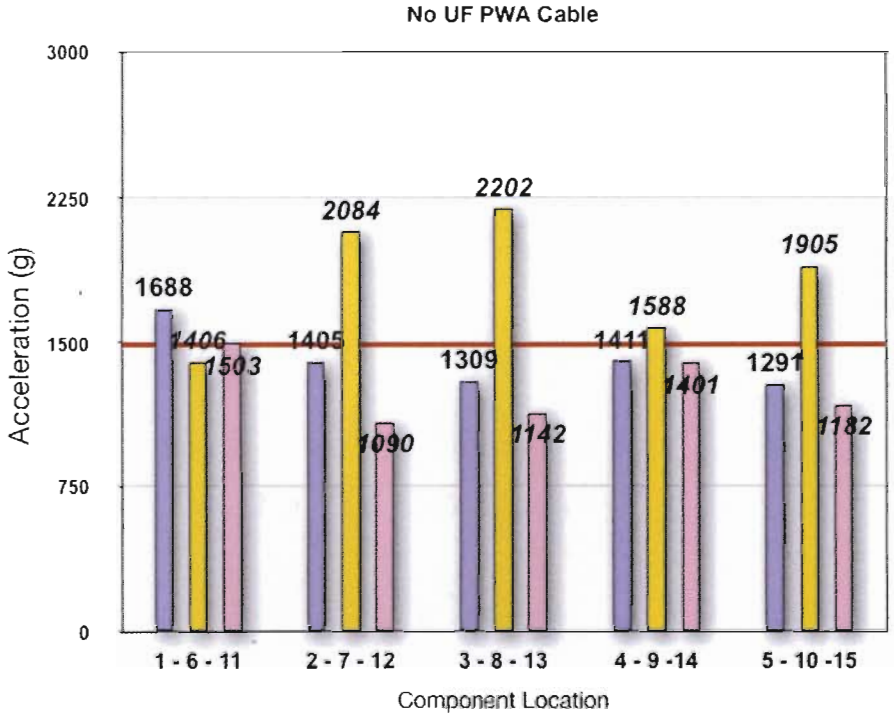

Figure 4.26: Accelerations on CSPs without edge-bond with DAQ cable

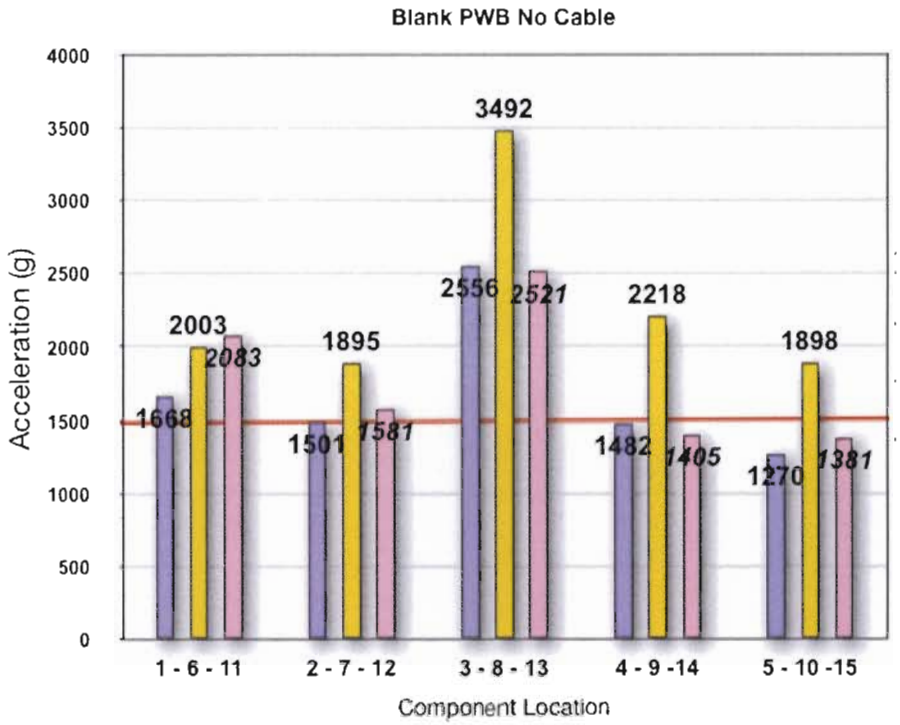

Figure 4.27: Accelerations on blank PWB without DAQ cable

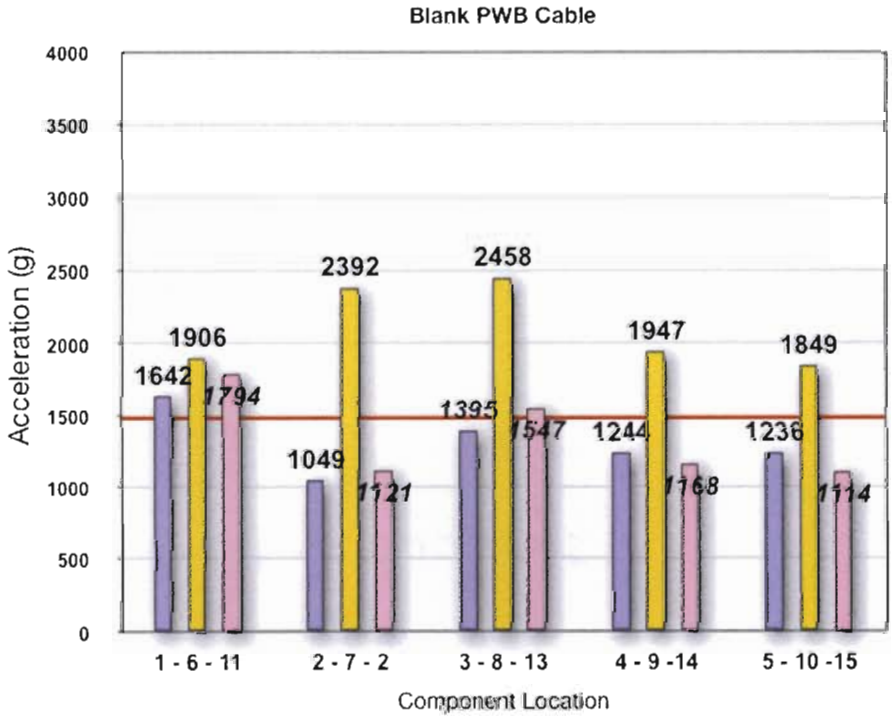

Figure 4.28: Accelerations on blank PWB without DAQ cable

Chapter 5

Conclusions

5.1 High-speed Data Acquisition System

The high-speed data acquisition system developed for this study was shown to be capable of in-situ detection of solder interconnect failures during drop impact testing. As such it is a useful tool in drop impact reliability research and testing. The software developed for this system has been released as open-source software in hopes that it may prove useful for other drop impact reliability researchers.

The DAQ system provides in-situ detection capability, however the cable appears to have an effect on the results. The result of observing the acceleration condition at each component location provides a preliminary explanation as to why the DAQ system tested components have different failure location sequences, and different lifetimes, than the post-drop detected components. It is expected that DAQ tested components would generally have detectable failure earlier than post-drop tested components due to the detection of transitional and intermittent failures. This was not found to be true, and preliminary analysis hints that the DAQ cable itself, by adding mass or damping effect to the test vehicle, changed the dynamic response of the test

vehicle enough to effect the component failure rates.

Future drop impact testing systems should minimize the chances of this influence by using lighter weight wires in the last few inches of cable before connecting to the test vehicle. It would also be preferable for the test vehicle to have a cable attached to both ends of the test vehicle to maintain symmetry between component locations due to the effects of the cable attachment, even if that effect is minimized. This would increase the damping effects of the cables, because there are two of them, but it would maintain the validity of grouping the component locations for analysis as suggested by the JEDEC standard [1].

Even with these caveats, data shows that high-speed data acquisition failure detection to be a more cost effective and reliable testing system than post-drop resistance measurement. There is great potential for a fully automated data acquisition based drop impact testing system. Automation would remove the need for an operator to manually conduct each drop test cycle, improve the overall accuracy of results, and reduce the chances of experimental error in resistance measurements.

5.2 CSP drop impact reliability

The drop test reliability of 0.5mm pitch CSPs assembled on the JEDEC test vehicle with lead-free SAC305 solder are studied by two failure detection systems. The following conclusions can be drawn from this research:

1. The component location plays a significant role in the drop test reliability. Generally speaking, components at the center of the board are more prone to fail due to higher strains. But the differences in drops-to-failure between the two different failure detection systems indicate that additional mass on the board, due to the DAQ system cable, changed the distribution of strains along the

board, which resulted in different failure locations.

2. Higher impact force or G-level resulted in lower drops-to-failure. But there are large variations in drops-to-failure between different boards under the same drop conditions.

3. Edge bonding can significantly improve drop test reliability for CSPs under JEDEC drop test conditions. The edge bonded CSPs typically survived 5 – 8 times longer at 2900G-0.3ms drop impacts, and 8 – 10 times longer at 1500G-0.5ms drop impacts.

4. The failure mode of the two different edge-bond materials are different when failure does occur. The acrylic material delaminates from the package sides, while the epoxy material fractures but maintains its adhesion to both board and CSP. The overall reliability improvement of increased drops-to-failure for both materials is similar.

5. The majority of drop test failures were due to traces breaking caused by cohesive failure of resin between the copper pads and the fiberglass dielectric layer. This indicates that the solder joints are not the weakest link area of the assembly in these drop impact conditions.

6. The pad cratering issue suggests that PWB laminate materials should be improved and the laminate specification and testing method should be included in a future JEDEC drop testing standard. At present, drop impact testing standards do not specifically address PWB pad cratering or resistance to this failure mode.

5.3 Future Work

Additional failure analysis of test vehicles is currently underway. That work is focused on gathering more information about PWB laminate material failure and the problem of pad cratering under drop impact conditions. This will hopefully lead to recommendations on PWB layout design rules to minimize effects of this problem, or recommendations on PWB laminate material improvement.

One of the goals of this project was to qualitatively analyze transitional solder interconnect failure due to drop impact. Transitional failure as used here is the condition where solder fracture has partially occurred but a portion of the solder interconnect remains unfractured. This is expected to cause minor resistance rise due to crack separation during the test vehicle deflection. The primary roadblock to achieving this result was the tendency of components to fail at the corner 2 I/O connections by trace breakage rather than solder interconnect fracture failure. Because only a few solder fractures were observed on the failed test vehicles it was not possible to draw conclusions on the ability of the high-speed data acquisition system to identify early transitional failure. A PWB layout targeted at preventing the I/O trace breakage failure may be necessary to further explore transitional solder fracture failure. In addition, the test vehicle must be more resilient to pad cratering to prevent trace breakage within the ball-grid array so that solder fracture failure is the most prevalent failure condition, which is currently not the case.

It is left to future work to verify that partial solder interconnect fractures can be detected by high-speed data acquisition such that a resistance change to crack area correlation is found. Although this study did identify several transitionally failed solder interconnects where an incomplete fracture was partially dyed red by the dye penetrant method, there were not enough similarly failed fractures to determine the resistance rise versus crack area correlation.

Bibliography

[1] JEDEC Standard JESD22-B111, "Board level drop test method of components for handheld electronic products", JEDEC Solid State Technology Assoc, 2003.

[2] JEDEC Standard JESD22-B104-C, "Mechanical Shock", JEDEC Solid State Technology Assoc, 2004.

[3] JEDEC Standard JESD22-B110A, "Subassembly Mechanical Shock", JEDEC Solid State Technology, 2004.

[4] C.T. Lim and Y.J. Low, "Investigating the drop impact of portable electronic products", Proceedings of 52nd IEEE Electronics Components and Technology Conference, 2002, pp. 1270–1274.

[5] C.T. Lim, C.W. Ang, L.B. Tan, S.K.W. Seah, and E.H. Wong, "Drop impact survey of portable electronic products", Proceedings of IEEE Electronics Components and Technology Conference, 2003, pp. 113–120.

[6] T. Y. Tee, H.S. Ng, C.T. Lim, E. Pek, and Z. Zhong, "Board level drop test and simulation of TFBGA packages for telecommunication applications", Proceedings of 53rd IEEE Electronics Components and Technology Conference, 2003, pp. 121–129.

[7] A. Syed, T.S. Kim, S.W. Cha, J. Scanlon, and C.G. Ryu, "Effect of Pb free alloy composition on drop/impact reliability of 0.4, 0.5 & 0.8mm pitch chip

scale packages with NiAu pad finish", Proceedings of 2007 IEEE Electronics Components and Technology Conference, pp. 951–956.

[8] A. Syed, T.S. Kim, Y.M. Cho, C.W. Kim, and M. Yoo, "Alloying effect of Ni, Co, and Sb in SAC solder for improved drop performance of chip scale packages with Cu OSP pad finish", Proceedings of 2006 IEEE Electronics Packaging Technology Conference, pp. 404–411.

[9] R.S. Pandher, B.G. Lewis, R. Vangaveti, and B. Singh, "Drop shock reliability of lead-free alloys – Effect of micro-additives", Proceedings of 2007 IEEE Electronics Components and Technology Conference, pp. 669–676.

[10] H. Kim, M. Zhang, C.M. Kumar, D. Suh, P. Liu, D. Kim, M. Xie, and Z. Wang, "Improved drop reliability performance with lead free solders of low Ag content and their failure modes", Proceedings of 2007 IEEE Electronics Components and Technology Conference, pp. 962–967.

[11] S. Zhang, C. Chen, and S. Yee, "An investigation on the reliability of CSP solder joints with numerous underfill materials", Journal of SMT, Vol. 16, No. 3, 2003, pp. 25–30.

[12] B. Toleno, D. Maslyk, and T. White, "Using underfills to enhance drop test reliability of Pb-free solder joints in advanced chip scale packages", Proceedings of 2007 SMTA Pan Pacific Symposium.

[13] Y. Liu, G. Tian, R.W. Johnson, and L. Crane, "Lead-free chip scale packages: assembly and drop test reliability", IEEE Transactions on Electronics Packaging Manufacturing, Vol. 29, No. 1, Jan. 2006, pp. 1–9.

[14] G. Tian, Y. Liu, R.W. Johnson, P. Lall, M. Palmer, M.N. Islam, and L. Crane, "Corner bonding of CSPs: processing and reliability", IEEE Transactions on Electronics Packaging Manufacturing, Vol. 28, No. 3, July 2005, pp. 231–240.

[15] H. Lou, X. Qu, Z. Chen, J. Wang, T. Lee, and H. Wang, "Lifetime assessment of solder joints of BGA package in board level drop test", Proceedings of IEEE 6th International Conf. on Electronic Packaging Technology, 2005, pp. 696–700.

[16] T.T. Mattila, P. Marjamaki, J.K. Kivilahti, "Reliability of CSP interconnections under mechanical shock loading conditions", IEEE Transactions on Components and Packaging Technologies, Vol. 29, No. 4, Dec. 2006, pp. 787–795.

[17] D.Y.R. Chong, K. Ng, J.Y.N. Tan, P.T.H. Low, J.H.L. Pang, F.X. Che, B.S. Xiong, L.H. Xu, "Drop impact reliability testing for lead-free & leaded soldered IC packages", Proceedings of 2005 IEEE Electronics Components and Technology Conference, pp. 622–629.

[18] T.Y. Tee, H.S. Ng, C.T. Lim, E. Pek, "Drop test and impact life prediction model for QFN packages", Journal of SMT, Vol. 16, No. 3, 2003, pp. 31–39.

[19] Y.S. Lai, P.F. Yang, and C.L. Yeh, "Experimental studies of board-level reliability of chip-scale packages subjected to JEDEC drop test condition", Microelectronics Reliability, Vol. 46, No. 2, 2006, pp. 645–650.

[20] P. Marjamaki, T. Mattila, J. Kivilahti, "Finite element analysis of lead-free drop test boards", Proceedings of 2005 IEEE Electronics Components and Technology Conference, pp. 462–466.

[21] J.E Luan, T.Y. Tee, E. Pek, C.T. Lim, Z. Zhong, J. Zhou, "Advanced numerical and experimental techniques for analysis of dynamic responses and solder joint reliability during drop impact", IEEE Transactions on Component and Packaging Technologies, Vol. 29, No. 3, Sept. 2006, pp. 449–456.

[22] F.X. Che, J.H.L. Pang, W.H. Zhu, W. Sun, A. Sun, C.K. Wang, H.B. Tan, "Comprehensive modeling of stress-strain behavior for lead-free solder joints un-

der board-level drop impact loading conditions", Proceedings of 2007 IEEE Electronics Components and Technology Conference, pp. 528–534.

[23] S. Park, C. Shah, J. Kwak, C. Jang, J. Pitarresi, T. Park, S. Jang, "Transient dynamic simulation and full-field test validation for a slim-PCB of mobile phone under drop/impact", Proceedings of 2007 IEEE Electronics Components and Technology Conference, pp.914–923.

[24] D.Y.R. Chong, F.X Che, J.H.L. Pang, L. Xu, B.S. Xiong, H.J. Toh, and B.K. Lim, "Evaluation of influencing factors of board-level drop reliability for chip scale packages (fine-pitch ball grid array)", IEEE Transactions on Advanced Packaging, 2007.

[25] E.H. Wong, R. Rajoo, Y.M. Mai, S.K.W. Seah, K.T. Tsai, and L.M. Yap, "Drop Impact: Fundamentals & Impact Characterization of Solder Joints", Proceedings of 2005 IEEE Electronics Components and Technology Conference, pp. 1202–1208.

Appendices

A Drop Testing Procedures

The procedures followed to drop test each test vehicle are listed below. These procedures were developed for use with the Lansmont M23 TTS II shock tester and Cal Poly's DropGather software controlled high-speed data acquisition system. The *drop test controller* is the touch-screen driven controller computer for the Lansmont M23 shock tester.

1. Prepare test vehicle for drop testing by recording board number, and soldering 16 wires into the through holes on the board.

2. Check DAQ cable wiring for correctness, and verify each solder connection has good continuity from the signal wire to the ground wire at the cable's connector end. Check that there are no bridged solder connections at the wires.

3. Select the intended drop impact pulse profile for this test vehicle. Set the Drop Height in the drop test controller for this drop impact pulse profile according to the earlier calibrated tests.

4. Set the impact surface for the intended drop impact pulse profile according to the earlier calibrated tests. Securely tape the strike surface into place.

5. Check or set Trigger Height to 1.5" and Brake Height to 1" in the drop test controller.

6. Check Auto Raise Table settings in the drop test controller. The table should auto raise after a drop is completed to optimize time spent testing.

7. Check the multiple drop cycling settings in the drop test controller. The drop cycle count should be set to 1 so that the controller never auto arms or drops the drop table before the DropGather software is ready to acquire sampled data during the test.

8. Test the drop impact pulse profile for accuracy before attaching the test vehicle to the drop table.

9. Attach test vehicle to the drop table using four screws into standoffs. Verify the board is not tightly bound between any two support screws but has slight freedom of motion in the X-Y plane of the test vehicle.

10. Route the DAQ cable behind the machine and away from the test vehicle. Clamp the cable to the drop table using the two #10-24 screws and cable clamps, leaving enough slack in the cable for the wires to loop away from the test vehicle gently. The wires must not apply any side-load to the test vehicle and restrict the freedom of motion in the X-Y plane of the test vehicle.

11. Connect the DAQ cable to the front of the NI connector boxes and secure the connector with the retention screws.

12. Check the cable connections from the NI connector boxes to the ADC in the computer and verify the cables are secured with the retention screws.

13. Turn on the DAQ power supply (if external power supply is in use). This is not necessary if the DAQ hardware is using the internal PCI power supply.

14. Open the index.ini DropGather configuration file and manually reset the drop number to zero.

15. Launch the DropGather software. Confirm the configuration variables including the board number, solder composition (lead or lead-free), surface finish (osp or enig), and test length (1fail or not) when prompted.

16. Confirm ready for test. Verify DropGather is now waiting for trigger signal.

17. Depress the manual trigger button on the connector box #1 to initiate the pre-testing sample run. The files for drop #0 should appear in the current working directory. If the output files are not created verify the cause of failure before beginning the drop test cycle for this test vehicle.

18. Reset the DropGather software for the next cycle by pressing Enter when prompted.

19. Start the Lansmont Test Partner software, verify an active connection to the signal conditioner. Check for the appropriate settings for the accelerometer currently connected to the signal conditioner.

20. Initiate the auto raise function on the drop test controller to raise the drop table to the Drop Height.

21. Execute the drop test run for this test vehicle by repeating the following steps for each test cycle until testing is complete:

    (a) Verify the DropGather software is waiting for trigger. If the software is not waiting for trigger correct this before continuing on and dropping the table.

    (b) Arm the Test Partner software to prepare it to record drop impact conditions.

    (c) Verify the Test Partner software is waiting for accelerometer signal triggering. If the software is not waiting for trigger correct this before continuing

on and dropping the table.

(d) Check that the safety floor mat is not locking-out the drop test controller.

(e) Arm the shock tester controller in preparation to drop the table.

(f) Press the Drop button on the drop test controller.

(g) Watch the drop table impact the table base and note any abnormal conditions such as the DAQ cable becoming trapped under the drop table. If there are any abnormal motions of the table or base during the drop test then stop the testing cycle until the reason is determined and corrected.

(h) Verify the DropGather software has collected sampled data. Review each of the result graphs individually as they are shown on-screen.

(i) After closing all result graphs write notes on the drop results, failures, or abnormal testing conditions for this test cycle.

(j) Verify the drop table has been auto-raised to the Drop Height by the drop test controller. Check for any components which have fallen off the test vehicle at this time. If any components have fallen off they must be collected and not allowed to remain under the test vehicle during testing since loose, bouncing components may cause extra damage to other components.

(k) Record the peak acceleration and pulse width as reported by the Test Partner software. If either result is out of reasonable error range for the current intended drop impact condition then make appropriate changes to the Drop Height or Strike Surface at this time.

(l) Confirm whether DropGather should repeat the currently configured test.

22. Conclude testing of this test vehicle:

(a) Cancel DropGather's request to repeat the currently configured test.

(b) Close the DropGather software console window.

(c) Turn off the DAQ power supply (if external power is in use). This is not necessary if the DAQ hardware is using the internal PCI power supply.

(d) Wait 10 seconds for complete discharge of the DAQ system.

(e) Disconnect the DAQ system cable from the connector boxes.

(f) Remove the test vehicle mounting screws from the drop table standoffs. Then unclamp the DAQ system cable from the drop table and remove the test vehicle from the drop table.

(g) Verify the integrity and continuity of the solder connections from the DAQ system cable to the test vehicle. If any solder connections are broken note the component and board number.

(h) Remove the DAQ system cable from the test vehicle by unsoldering the wires in the through holes.

B DAQ System Observed Failure Conditions

The DAQ system was designed to detect transitional failure in solder interconnections as described in Section 2.2. Figures B-1, B-2, and B-3 are examples of the output the DAQ system generates for each of the following failure conditions:

1. Transitional Failure – defined as a small rise in resistance during drop impact testing with no post-drop detectable change in resistance.

2. Intermittent Failure – defined as a significant resistance change (more than 100Ω) or discontinuity during drop impact testing which is no longer significant after the test vehicle comes to rest. This may be post-drop detectable as a small resistance change or may not be post-drop detectable at all.

3. Complete Failure – defined as a permanent discontinuity or significant resistance change (more than 100Ω) which is present after the test vehicle comes to rest. This condition should always be post-drop detectable.

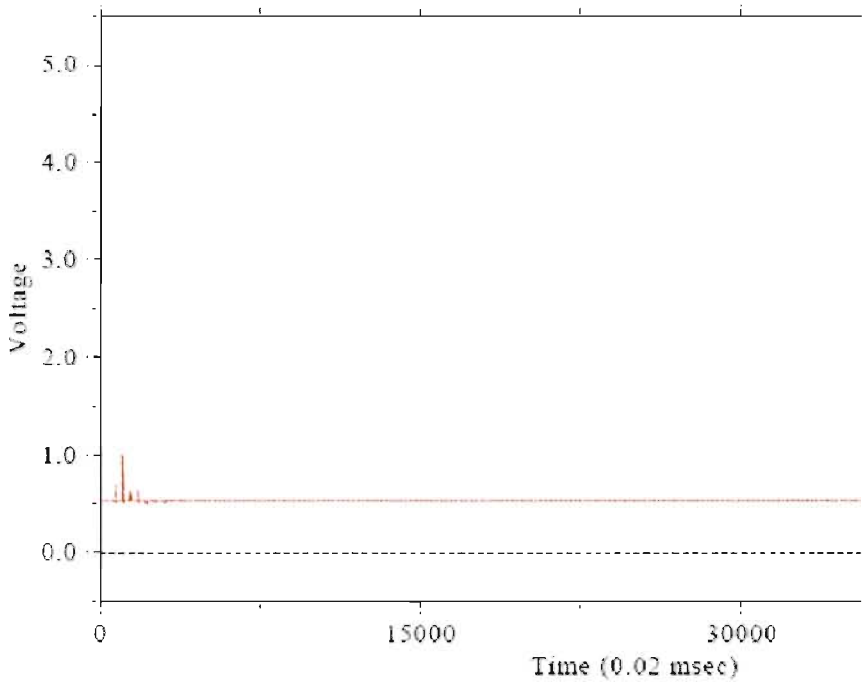

Figure B-1: DAQ output of sampled Transitional Failure condition

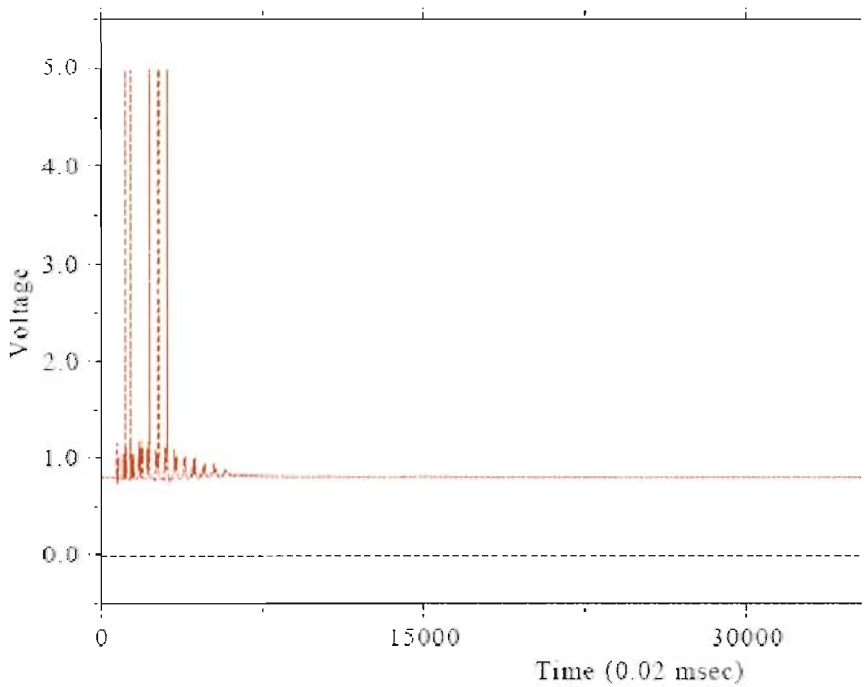

Figure B-2: DAQ output of sampled Intermittent Failure condition

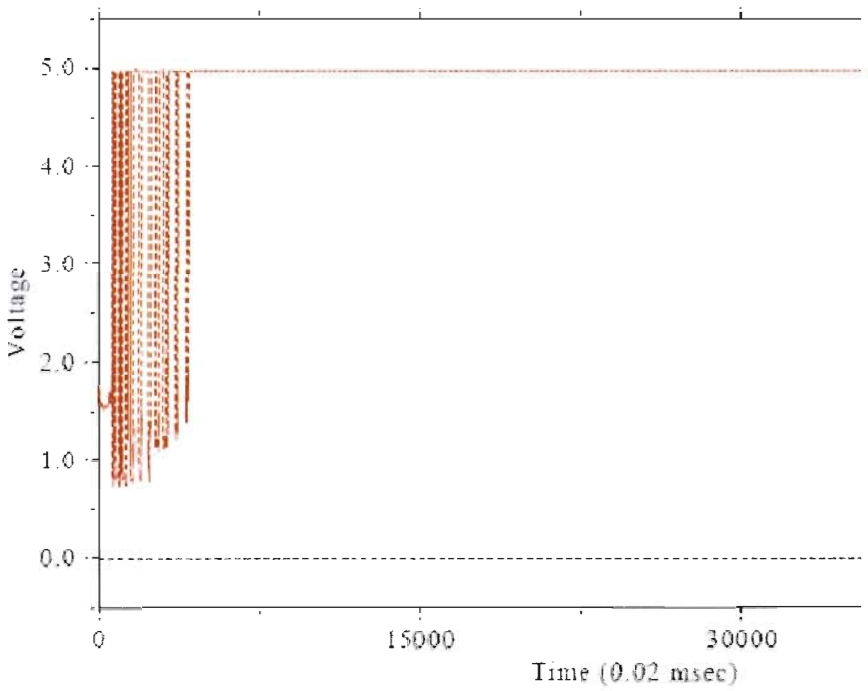

Figure B-3: DAQ output of sampled Complete Failure condition

C DropGather Software C++ Source Code

file: readme.txt

```
DropGather README
last updated - Fall 2007

maintainer: Andrew Farris <ajfarris@gmail.com>
website: http://www.lordmorgul.net/dropgather

***
This file is part of DropGather.

DropGather is free software: you can redistribute it and/or
    modify
it under the terms of the GNU General Public License as
    published by
the Free Software Foundation, either version 3 of the
    License, or
(at your option) any later version.

DropGather is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty
    of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public
    License
along with DropGather.  If not, see
    <http://www.gnu.org/licenses/>.
***

DropGather is a control and output program for the high-speed
     analog-to-digital sampling data acquisition system
    developed for research into electronics drop testing
    reliability.  The program controls the National
    Instruments (NI) ADC to collect data from the NI connector
     box housed analog inputs.  The NI connector boxes have an
     internally wired voltage divider circuit connected to the
     chips to be drop tested.  The software controls the
    sampling start time, total samples taken for each channel,
     and channel configuration.

While DropGather is open-source software it currently depends
     on non-open libraries to be useful.  These are the
    National Instruments DAQmx hardware driver library and
    also the DISLIN scientific data plotting library.

For Documentation on the design of DropGather see the thesis
    paper by Andrew Farris posted at the above mentioned
    DropGather website.  Additional documentation is not yet
    available but could be provided upon request.  If you
    would like to use or modify DropGather for your own
```

research project please feel free to contact Andrew.

CHANGELOG:

Fall 2007: Andrew Farris

- remove operator comments per drop cycle for better workflow
- change to default yes response from operator to repeat drop cycle

Spring 2007: Andrew Farris

- changed voltage range for trigger plots to 10v (resistor chosen for trigger)
- changed voltage range for chip signal plots to 5.5v (clearly show high at 5v)
- combined voltage and trigger plots for faster visual inspection (not in pdf)
- separated chip signal plots for clarity (one per window) and faster testing

Winter 2007: Andrew Farris

- removed sampling of accelerometer input, added trigger
- added default configuration flags to bypass user input for underfill, solder, and coating
- changed channel input ranges for new connector box
  configured input channels are:
  box 1, trigger: dev1/ai0
  box 1, chips 1-8: dev1/ai1:8
  box 2, 5v rail: dev1/ai16

box 2, chips 9-15: dev1/ai17:23

box 2, 5v rail: dev1/ai24

- input ranges in use: dev1/ai0:8;dev1/ai17:24
- leaving the middle rail signal unused, easier to keep the chip signals together in the data block

Fall 2006: Keith Rothman

- original software design and implementation

FILES:

taskhandler.h	– wrapper on DAQmx task control functions
errorhandlers.h	– wrapper on DAQmx error function output
DataCollection.h	– wrapper for sampled data from ADC
imeProject.h	– functions for the running sampling task
dropgather.cpp	– main and operator configuration options
imeProject.cpp	– provides most of the worker functions for the software, data handling/output
DAQError.cpp	– wrapper definition for DAQmx exception

file – taskhandler.h

```
/*  This file is part of DropGather.

  DropGather is free software: you can redistribute it and/or
     modify
  it under the terms of the GNU General Public License as
     published by
  the Free Software Foundation, either version 3 of the License,
     or
  (at your option) any later version.

  DropGather is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public
     License
  along with DropGather.  If not, see
     <http://www.gnu.org/licenses/>.
*/
#include "NIDAQmx.h"
#include "errorhandlers.h"

#ifndef TASKHANDLER
#define TASKHANDLER

class TaskHandler
{
  TaskHandler(TaskHandler & copy) {};
  TaskHandler operator = (TaskHandler & copy) {};
  TaskHandle  taskHandle;
  std::string name;
public:
  TaskHandler() : taskHandle(0), name("")
  {
    char cname[100];
    DAQError(DAQmxCreateTask("",&taskHandle));
    DAQError(DAQmxGetTaskName(taskHandle, cname, 100));
    name = cname;
  };

  TaskHandler(const char * cname)
  {
    char cname2[100];
    DAQError(DAQmxCreateTask(cname,&taskHandle));
    DAQError(DAQmxGetTaskName(taskHandle, cname2, 100));
```

```
    name = cname2;
  };

  operator TaskHandle()
  {
    return taskHandle;
  }

  ~TaskHandler()
  {
    if( taskHandle!=0 ) {
    /*********************************************/
    // DAQmx Stop Code
    /*********************************************/
    DAQmxStopTask(taskHandle);
    DAQmxClearTask(taskHandle);
    }
  }
};
#endif
```

file – errorhandlers.h

```
/*  This file is part of DropGather.

  DropGather is free software: you can redistribute it and/or
     modify
  it under the terms of the GNU General Public License as
     published by
  the Free Software Foundation, either version 3 of the License,
     or
  (at your option) any later version.

  DropGather is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public
     License
  along with DropGather.  If not, see
     <http://www.gnu.org/licenses/>.
*/
#include <exception>
#include <string>

#ifndef EXCPT
#define EXCPT

class Excpt : public std::exception
{
  Excpt();
public:
  Excpt(int _error, std::string _emsg) throw() : error(_error),
     emsg(_emsg) {};

  const char * what() throw()
  {
    return emsg.c_str();
  };

    int error;
  std::string emsg;
};

void DAQError(int32 error);

#endif
```

file – DataCollection.h

```
/*  This file is part of DropGather.

  DropGather is free software: you can redistribute it and/or
     modify
  it under the terms of the GNU General Public License as
     published by
  the Free Software Foundation, either version 3 of the License,
     or
  (at your option) any later version.

  DropGather is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public
     License
  along with DropGather.  If not, see
     <http://www.gnu.org/licenses/>.
*/
#include "NIDAQmx.h"
#include "errorhandlers.h"

#ifndef DATACOLLECTION
#define DATACOLLECTION

class DataCollection
{
  int nChans;
  int nSamples;
  float64 * data;
  DataCollection() {};
public:
  DataCollection(int _nChans, int _nSamples) : nChans(_nChans),
     nSamples(_nSamples)
  {
    data = new float64[nSamples*nChans];
  };

  ~DataCollection()
  {
    delete [] data;
  };

  operator float64*()
  {
```

```
    return data;
  }

  float64 ReadVal(int Chan, int Sample)
  {
    return data[Chan*nSamples+Sample];
  }

  void GetChan(int Chan, float * Samples, int _nSamples)
  {
    if(Chan >= nChans)
      throw Excpt(0, "Index channel above");

    if(nSamples < _nSamples)
      _nSamples = nSamples;

    for(int i = 0; i < _nSamples; i++)
      Samples[i] = (float) data[Chan*nSamples + i];
  }

  void GetChan(int Chan, float64 * Samples, int _nSamples)
  {
    if(Chan >= nChans)
      throw Excpt(0, "Index channel above");

    if(nSamples < _nSamples)
      _nSamples = nSamples;

    for(int i = 0; i < _nSamples; i++)
      Samples[i] = data[Chan*nSamples + i];
  }
};

#endif
```

file – imeProject.h

```
/*  This file is part of DropGather.

  DropGather is free software: you can redistribute it and/or
     modify
  it under the terms of the GNU General Public License as
     published by
  the Free Software Foundation, either version 3 of the License,
     or
  (at your option) any later version.

  DropGather is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public
     License
  along with DropGather.  If not, see
     <http://www.gnu.org/licenses/>.
*/
#include "DataCollection.h"
#include "taskhandler.h"
#include <fstream>
#include <iostream>
#include <sstream>
#include <vector>
#include "dislin.h"

#ifndef IMEPROJ
#define IMEPROJ

class imeProject
{
  TaskHandler  taskHandle;
  int32        read;
  const int nSamples;
  const int nChans;
  int drop;
  DataCollection data;
  float * chan;
  float * tarray;

  imeProject() : nSamples(0), nChans(0), data(0,0) {};

  std::fstream db;
  std::fstream datafile;
```

```
  std::stringstream proc;
public:
  imeProject(int, int);
  ~imeProject();
  void doCase(bool fail1, bool PbAg, bool Fill, bool Finish, int
     index);
  void imeProject::showCurveWnd(const int index);
};

#endif //IMEPROJ
```

file – dropgather.cpp

```
/*  This file is part of DropGather.

  DropGather is free software: you can redistribute it and/or
     modify
  it under the terms of the GNU General Public License as
     published by
  the Free Software Foundation, either version 3 of the License,
     or
  (at your option) any later version.

  DropGather is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public
     License
  along with DropGather.  If not, see
     <http://www.gnu.org/licenses/>.
*/
#include "imeProject.h"
#include <iostream>
#include <conio.h>

int main(int argc, char * argv)
{
  std::stringstream proc;
  std::string input;
  bool fail1 = false;
  bool PbAg = false;
  bool Fill = false;
  bool Finish = false;  // osp or enig, osp is false
  bool rockandroll = false;

  bool cfgfail = false;
  bool cfgsolder = false;
  bool cfgfill = false;
  bool cfgfinish = false;

   // TODO make CLI parameters for these
   bool preconf = false;
   int boardnum = 0;

   const int chancount = 17;
   const int samplecount = 50000;  // approx:  1 s = 50000, 2s =
      100000, 0.5s = 25000
```

```
// preconfiguring
cfgfail = true; // leave not 1fail
cfgsolder = true; // leave not lead
Finish = true; // osp
cfgfinish = true; // leave all osp

/* setup new configuration questions and base filename config,
    using only
board number and underfill flag

base filename should be used with drop counter combined later
   at file creation
*/

try
{
  bool quit = false;
  imeProject proj(chancount, samplecount); // Number of
     channels, number of samples

  while(!quit)
  {
    while(!rockandroll)
    {
      while(!cfgfail) // 1fail
      {
        std::cout << "Is this board marked 1-Fail? Y/[N]" <<
           std::endl;
        std::getline(std::cin, input);
        if(input == "y" || input == "Y")
        {
          cfgfail = true;
          fail1 = true;
          break;
        }
        else if(input == "n" || input == "N" || input == "")
        {
          cfgfail = true;
          fail1 = false;
          break;
        }
        else
        {
          std::cout << "I didn't understand your response, try
             again." << std::endl;
```

```
    continue;
  }
}

while(!cfgsolder)
{
  std::cout << "Is this board lead? Y/[N]" << std::endl;
  std::getline(std::cin, input);
  if(input == "y" || input == "Y")
  {
    cfgsolder = true;
    PbAg = true;
    break;
  }
  else if(input == "n" || input == "N" || input == "")
  {
    cfgsolder = true;
    PbAg = false;
    break;
  }
  else
  {
    std::cout << "I didn't understand your response, try
       again." << std::endl;
    continue;
  }
}

while(!cfgfill)
{
  std::cout << "Is this board underfilled? Y/[N]" <<
     std::endl;
  std::getline(std::cin, input);
  if(input == "y" || input == "Y")
  {
    cfgfill = true;
    Fill = true;
    break;
  }
  else if(input == "n" || input == "N" || input == "")
  {
    cfgfill = true;
    Fill = false;
    break;
  }
  else
  {
```

```
    std::cout << "I didn't understand your response, try
        again." << std::endl;
    continue;
  }
}

while(!cfgfinish)
{
  std::cout << "Is this board finish OSP? Y/[N]" <<
     std::endl;
  std::getline(std::cin, input);
  if(input == "y" || input == "Y")
  {
    cfgfinish = true;
    Finish = true;
    break;
  }
  else if(input == "n" || input == "N" || input == "")
  {
    cfgfinish = true;
    Finish = false;
    break;
  }
  else
  {
    std::cout << "I didn't understand your response, try
        again." << std::endl;
    continue;
  }
}

while(!boardnum) // dont ask about board number if it is
   already set
{
  std::cout << "What is the board number?" << std::endl;
  std::getline(std::cin, input);
  proc.clear();
  proc.str("");
  proc.str(input);

  proc >> boardnum;

  if(proc.fail())
  {
    std::cout << "I didn't understand your response, try
        again." << std::endl;
    continue;
```

```
  }
  break;
};

std::cout << "Board is one fail: ";
if(fail1)
{
  std::cout << "Yes";
}
else
{
  std::cout << "No";
}
std::cout << std::endl;

std::cout << "Solder paste is:    ";
if(PbAg)
  std::cout << "Pb";
else
  std::cout << "Lf";
std::cout << std::endl;

std::cout << "Board is filled:    ";
if(Fill)
  std::cout << "Yes";
else
  std::cout << "No";
std::cout << std::endl;

std::cout << "Board finish is:    ";
if(Finish)
  std::cout << "OSP";
else
  std::cout << "ENIG";
std::cout << std::endl;

std::cout << "Board number is:      ";
std::cout << boardnum << std::endl << std::endl;

while(1)
{
  std::cout << "Is everything correct? Y/[N]" <<
      std::endl;
  std::getline(std::cin, input);
  if(input == "y" || input == "Y")
  {
    rockandroll = true;
```

```
        break;
      }
      else if(input == "n" || input == "N" || input == "")
      {
        rockandroll = false;
              // unset all preconf flags
              cfgfail = cfgsolder = cfgfill = cfgfinish =
                 false; // requery all config
              boardnum = 0; // cause requery for board number
        break;
      }
      else
      {
        std::cout << "I didn't understand your response, try
           again." << std::endl;
        continue;
      }
    }
  }

     /* run the configured cycle */
  proj.doCase(fail1, PbAg, Fill, Finish, boardnum);
  rockandroll = false;


  while(1)
  {
    std::cout << "Is there another case? [Y]/N" << std::endl;
    std::getline(std::cin, input);
    if(input == "y" || input == "Y")
    {
      quit = false;
      break;
    }
    else if(input == "n" || input == "N" || input == "")
    {
      quit = true;
      break;
    }
    else
    {
      std::cout << "I didn't understand your response, try
         again." << std::endl;
      continue;
    }
  }
}
```

```
  } catch (Excpt & error)
  {
    std::cout << error.emsg << std::endl;
  }

  std::cout << "Exiting..." << std::endl;

  return 0;
}
```

file – imeProject.cpp

```
/*  This file is part of DropGather.

  DropGather is free software: you can redistribute it and/or
     modify
  it under the terms of the GNU General Public License as
     published by
  the Free Software Foundation, either version 3 of the License,
     or
  (at your option) any later version.

  DropGather is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public
     License
  along with DropGather.  If not, see
     <http://www.gnu.org/licenses/>.
*/
#include "imeProject.h"

imeProject::imeProject(int _nChans, int _nSamples) :
   nSamples(_nSamples), nChans(_nChans), data(nChans, nSamples)
{
  chan = new float[_nSamples];
  tarray = new float[_nSamples];
  for(int i = 0; i < _nSamples; i++)
    tarray[i] = (float) i;

  db.open("index.ini",std::ios_base::in);
  if(db.is_open())
  {
    db >> drop;
    if(db.fail() && !db.eof())
      throw Excpt(0, "Index not right");
  }
  else
       drop = 0;
  db.close();

  /*********************************************/
  // DAQmx Configure Code
  /*********************************************/
    // analog channels used are:
    // ai0     trigger          box1
```

```
    // ai1-7  chips 1-8 (8 of them) box1
    // ai16   signal voltage (5v)   box2 - ai0
    // ai17-23  chips 9-15 (7 of them)  box2 - ai1-7
    // ai24   signal voltage (5v)   box2 - ai8
    // the signal voltage is connected to two inputs, only one
       needs to be used
    // software assumes ai24 is used right now.. output formats
       have to change to use ai16
  DAQError (DAQmxCreateAIVoltageChan(
    taskHandle,    // task to append too
    "Dev1/ai0:8,Dev1/ai17:24",  // Channels to append
    "",        // Channel names, defaults to ai0 ai1 ... ai16
    DAQmx_Val_RSE,  // Referenced single ended
    0.0,       // Minimum voltage to see
    10.0,      // Maximum voltage
    DAQmx_Val_Volts,// Scale to use
    NULL));      // Name of scale if custom

  DAQError (DAQmxCfgSampClkTiming(
    taskHandle,        // task to append too
    NULL,          // Timing mechinism
    50000.0,         // Samples per second
    DAQmx_Val_Rising,    // Edge to collect samples
    DAQmx_Val_FiniteSamps,  // Type of collection
    nSamples));        // Ammount of samples to collect

  DAQError (DAQmxCfgAnlgEdgeStartTrig(
    taskHandle,        // task to append too
    "Dev1/ai0",      // Name of channel to trigger on
    DAQmx_Val_RisingSlope,     // Edge to detect
    1.5));           // Trigger level
  DAQError (DAQmxSetAnlgEdgeStartTrigHyst(taskHandle, 0.1));
};

void imeProject::doCase(bool fail1, bool PbAg, bool Fill, bool
    Finish, int index)
{
  std::string pdffile;
  proc.str("");

  proc << drop << "_";

  if(fail1)
    proc << "1Fail_";

  if(PbAg)
    proc << "Pb_";
```

```
else
  proc << "Ag_";

if(Fill)
  proc << "UF_";
else
  proc << "NoUF_";

if(Finish)
  proc << "OSP_";
else
  proc << "ENIG_";

proc << "board_";
proc << index;
pdffile = proc.str();
proc << ".txt";

datafile.open(proc.str().c_str(),std::ios_base::out &
   std::ios_base::ate);

if(datafile.is_open())
{
  throw Excpt(0, "File already exists!");
}
else
{
  datafile.close();
  datafile.clear();
  datafile.open(proc.str().c_str(),std::ios_base::out);
}

if(!datafile.is_open())
  throw Excpt(0, "File couldn't be created");

/*********************************************/
// DAQmx Start Code
/*********************************************/
DAQError (DAQmxStartTask(taskHandle));

std::cout << " Ready... Waiting for trigger" << std::endl;

/*********************************************/
// DAQmx Read Code
/*********************************************/
DAQError (DAQmxReadAnalogF64(
  taskHandle,          // task to read from
```

```
    DAQmx_Val_Auto,        // How many samples to read
    DAQmx_Val_WaitInfinitely, // How long to wait for data
    DAQmx_Val_GroupByChannel, // How to fill array
    data,            // Array to be filled
    nChans*nSamples,       // Size of array to be filled
    &read,             // How many samples read
    NULL));            // Reserved, pass NULL, unknown function

  std::cout << read << " samples read." << std::endl;

  DAQError(DAQmxStopTask(taskHandle));


/*********************************/
// graphical display begins
  metafl("xwin");
    setpag("da4l");
  winsiz(800,600);

// graph trigger and rail voltage together
  disini();
  pagera();
  hwfont();
  axspos(450,1800);
  axslen(2200,1200);

  name("Time (0.02 msec)","x");
  name("Voltage","y");

  labdig(-1,"x");
  titlin("Data",1);
  graf(0.f, (float)nSamples, 0.f, 15000.f, -.5f, 10.f, 0.f, 1.f);
  title();

  char legendb[160];
  legini(legendb, 2, 20);
  leglin(legendb, "Trigger", 1);
  leglin(legendb, "5v Source", 2);

  color("magenta");
  // trigger data
  data.GetChan(0, chan, read);
  curve(tarray,chan,read);
  color("cyan");
  // rail voltage data
  data.GetChan(16, chan, read);
  curve(tarray,chan,read);
```

```
  color("fore");
  dash();
  xaxgit();
  legend(legendb, 7);
  disfin();

  // graph separate data channels, all 15 in their own window
  for (int ndx = 1; ndx < 16; ndx++)
  {
    showCurveWnd(ndx);
  }


/*********************************/
// PDF output file begins
  std::cout << "Writing PDF...";

/* buffer standard out */
//    std::cout.

  pdffile += ".pdf";
  char * file;
  file = new char[pdffile.size()+1];
  strcpy(file,pdffile.c_str());
    setfil(file);
  metafl("pdf");

    //setpag("da4l");
  disini();

  char *titles[] = {"Trigger","Chip 1","Chip 2","Chip 3","Chip
     4","Chip 5","Chip 6","Chip 7","Chip 8","Chip 9","Chip
     10","Chip 11","Chip 12","Chip 13","Chip 14","Chip 15","5v"};

  // graph trigger separately due to range issue
  int i = 0;
  pagera();
  hwfont();
  axspos(450,1800);
  axslen(2200,1200);

  name("Time (0.02 msec)","x");
  name("Voltage","y");

  labdig(-1,"x");
```

```
titlin(titles[i],1);

graf(0.f, (float)nSamples, 0.f, 15000.f, -.5f, 10.f, 0.f, 1.f);
// x-lower x-upper first-x x-step y-lower y-upper first-y
   y-step
title();

  color("red");
data.GetChan(i, chan, read);
curve(tarray,chan,read);

color("fore");
dash();
xaxgit();

endgrf();
newpag();

// continue graphing all other channels into the PDF
for(int i = 1; i < nChans; i++)
{
  pagera();
  hwfont();
  axspos(450,1800);
  axslen(2200,1200);

  name("Time (0.02 msec)","x");
  name("Voltage","y");

  labdig(-1,"x");

  titlin(titles[i],1);

  graf(0.f, (float)nSamples, 0.f, 15000.f, -.5f, 5.5f, 0.f,
     1.f);
  // x-lower x-upper first-x x-step y-lower y-upper first-y
     y-step
  title();

      color("red");
  data.GetChan(i, chan, read);
  curve(tarray,chan,read);

  color("fore");
  dash();
  xaxgit();
```

```
    endgrf();
    if(i != nChans-1)
    {
      newpag();
    }
  }
  disfin();
  delete [] file;

  std::cout << "Done" << std::endl;


/*********************************/
// text data file output begins
//  std::cout << "List any new damage to board or any components
   that have failed" << std::endl;
//  std::cout << "Limit your response to less than 20 lines" <<
   std::endl;
//  std::cout << "Type 'eof' followed by the <enter> key to end
   text entry" << std::endl << std::endl;
  std::vector<std::string> lines;
  std::string oneline = "";

  // removed commenting block to speedup working
/*
  while(oneline != "eof" && lines.size() < 21)
  {
    std::getline(std::cin, oneline);
    lines.push_back(oneline);
  }
*/
  std::cout << "Writing to file..." << std::endl;

  datafile << "usr" << std::endl;
/*
  for(size_t i = 0; i < lines.size()-1; i++)
  {
    datafile << lines[i] << std::endl;
  }
*/
  // changed index from lines.size()-1 to this as it is, removed
     comment handling loop above
/*
  for(size_t i = lines.size(); i < 20; i++)
  {
    datafile << std::endl;
  }
```

```
*/
  datafile << read << " " << nChans << std::endl;

  datafile << "Trigger\tChip 1\tChip 2\tChip 3\tChip 4\tChip
     5\tChip 6\tChip 7\tChip 8\tChip 9\tChip 10\tChip 11\tChip
     12\tChip 13\tChip 14\tChip 15\t5v" << std::endl;

  datafile.precision(6);
  datafile.fill(0);
  datafile.width(8);
  datafile.setf(datafile.fixed|datafile.left);

    for(int i = 0; i < nSamples; i++)
  {
    datafile << data.ReadVal(0, i);
    for(int j = 1; j < nChans; j++)
      datafile << "\t" << data.ReadVal(j, i);
    datafile << "\n";
  }

  // Get user description
  if(datafile.eof())
    throw Excpt(0, "File error!");

  datafile.close();

  std::cout << "Drop index " << drop << " completed." <<
     std::endl;

  drop++;

  // attempt to dump drop count to index file every time
  // no warning on fail here, leave file output failure until
     program exit
  db.clear();
  db.open("index.ini",std::ios_base::out);
  if(db.is_open())
  {
    db << drop;
    db.close();
  }
};

imeProject::~imeProject()
{
  delete [] tarray;
  delete [] chan;
```

```
  db.clear();
  db.open("index.ini",std::ios_base::out);
  if(db.is_open())
  {
    db << drop;
    if(db.fail() && !db.eof())
      throw Excpt(0, "Index not written check");
  }
};

// shows a plot in window for the data indicated by chan
// input:
// part - string, the part number or other label
// index - int, channel index for the call to GetChan()
void imeProject::showCurveWnd(const int index)
{
  disini();
  pagera();
  hwfont();
  axspos(450,1800);
  axslen(2200,1200);

  name("Time (0.02 msec)","x");
  name("Voltage","y");

  labdig(-1,"x");
  titlin("Data",1);
  graf(0.f, (float)nSamples, 0.f, 15000.f, -.5f, 5.5f, 0.f, 1.f);
  title();

  char legendb[160];
  char legtext[10];
  sprintf(legtext, "Part %d\0", index);
  legini(legendb, 1, 20);
  leglin(legendb, legtext, 1);
  char* colors[7] = {"green", "yellow", "magenta", "cyan", "red",
      "blue", "orange"};
  color(colors[index%2]); // alternate green and yellow
  data.GetChan(index, chan, read);
  curve(tarray,chan,read);

  color("fore");
  dash();
  xaxgit();
  legend(legendb, 7); // display upper right
  disfin();
}
```

file – DAQError.cpp

```
/*  This file is part of DropGather.

  DropGather is free software: you can redistribute it and/or
     modify
  it under the terms of the GNU General Public License as
     published by
  the Free Software Foundation, either version 3 of the License,
     or
  (at your option) any later version.

  DropGather is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public
     License
  along with DropGather.  If not, see
     <http://www.gnu.org/licenses/>.
*/
#include <NIDAQmx.h>
#include "errorhandlers.h"

void DAQError(int32 error)
{
  if(error != 0)
  {
    char * msg;
    int32 size = DAQmxGetErrorString(error,NULL,0);
    if(size > 0)
    {
            msg = new char[size];
    }
    else
    {
      msg = new char[512];
      size = 512;
    }
    DAQmxGetErrorString(error, msg, size);
    Excpt expt(error, msg);
    delete [] msg;
    throw expt;
  }
};
```